UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Evaluación de la popularidad en los sistemas de recomendación

Doble Máster en Ingeniería Informática e Investigación e Innovación en Tecnologías de la Información y las Comunicaciones

Autor: Cañamares Pérez, Rocío
Tutor: Castells Azpilicueta, Pablo

FECHA: Julio, 2016

Resumen

Los sistemas de recomendación buscan proponer por iniciativa propia a los usuarios opciones que probablemente aporten un valor a éstos. Implícita en la propia noción de recomendación está generalmente la idea de que cada persona puede recibir sugerencias personalizadas a sus necesidades y gustos particulares, por lo que cabría imaginar que los algoritmos de recomendación más eficaces deberían ser muy personalizados. Sin embargo, se ha observado en años recientes que la recomendación por mayorías (popularidad) no resulta mucho peor que los mejores algoritmos, y más aún, estos últimos tienden a sesgar sus recomendaciones hacia opciones mayoritarias. Sin embargo es un hecho conocido, aunque aún no estudiado en profundidad, que las metodologías actuales de evaluación priman por diseño los algoritmos que se sesguen hacia las recomendaciones populares. Es relevante, por tanto, entender en qué medida y bajo qué circunstancias la popularidad es o no un ingrediente realmente eficaz en la recomendación, y si en su caso lo es o no por efecto de un sesgo en los métodos de evaluación, pues de esta comprensión se pueden derivar conclusiones acerca de la efectividad de otros muchos algoritmos que se encuentran influenciados por la distribución de popularidad.

El presente trabajo aborda esta cuestión a nivel tanto teórico como empírico. En la vertiente teórica, desarrollamos una formulación analítica de la efectividad de la recomendación por popularidad en base a modelos probabilísticos, distinguiendo entre la efectividad observada (la que habitualmente se obtiene en los experimentos que se realizan en el área) y la real (la que se podría medir idealmente si se dispusiera de información exhaustiva de los gustos de los usuarios). A partir de esta estudiamos la influencia en dicha efectividad de distintos aspectos, como el tipo de partición de datos (en entrenamiento y test), la distribución de popularidad, la de descubrimiento, los gustos del usuario o su comportamiento a la hora de decidir sobre lo que votar o no.

Junto con el estudio analítico, llevamos a cabo un contraste empírico de hipótesis, empleando para ello conjuntos de prueba provenientes tanto de datos reales como de simulaciones. A fin de poder contrastar mediciones entre lo observado y lo real, realizamos un experimento con usuarios reales en una plataforma de crowdsourcing que nos permite, por un lado, obtener un conjunto de preferencias en ausencia de sesgos de descubrimiento y, por otro, reproducir y analizar la diferencia entre resultados observables y reales.

Entre los principales hallazgos derivados del estudio destaca la constatación de la existencia de situaciones en las que la recomendación por popularidad es contraproducente hasta el punto de ser peor que la recomendación aleatoria; así como situaciones donde la efectividad observada de la recomendación observada presenta contradicciones con la efectividad real. A nivel general, el estudio proporciona una identificación y comprensión de factores fundamentales que determinan estas situaciones.

Palabras clave: recomendación, popularidad, evaluación, efectividad observada, efectividad real, relevancia, descubrimiento.

Abstract

Recommender systems aim to suggest, on their own initiative, choices the user may find interesting or useful. Implicit in the concept of recommendation is the idea that each user may draw further benefit from a personalized recommendation that is tailored to her individual personal preferences, so it is reasonable to expect that highly personalized algorithms should be the most effective. However, it has been recently observed that recommending the most popular items is not as worse a strategy as one would expect than the best and more sophisticated personalized recommendation algorithms in the state of the art. Moreover, a bias towards popularity is, in fact, present in the best-considered algorithms. On the other hand, it is a well-known (though not deeply studied yet) fact that current offline evaluation methodologies reward algorithms that are biased towards popular recommendations. Considering all this, it would be important to understand the circumstances which make the recommendation by popularity an effective approach.

The present work address this question both at the theoretical and empirical levels. On the theoretical side, we develop an analytical formulation for popularity effectiveness in terms of probabilistic models. We distinguish observed effectiveness (the one that is usually obtained in common experiments and is based on observed data) and real effectiveness (the one that we could ideally measure if we had complete information of user preferences). Using these expressions we are able to study the influence of different aspects in the outcome of an experiment: the split procedure, the popularity and discovery distributions, the preferences of users and their behavior when faced to rating decisions.

Along with the formal study, we carry out an empirical analysis to confirm theoretical conclusions, using both real datasets and simulations. With the objective to compare observed and real measures, we develop an experiment with real users from a crowdsourcing platform. This experiment allows us to obtain an unbiased observation of user preferences, and study the difference between observed and real effectiveness.

Among the main findings derived from the theoretical and empirical analysis we confirm the existence of situations in which popularity is worse than random recommendation, as well as situations in which the observed and real effectiveness disagree. More generally, the study provides an identification and understanding of fundamental factors that determine these and other situations.

Key words: recommendation, popularity, evaluation, effectiveness, observed effectiveness, real effectiveness, relevance, discovery.

Agradecimientos

En primer lugar quería dar las gracias al grupo de investigación IRG (Information Retrieval Group) por darme la oportunidad de llevar a cabo este proyecto. En concreto, quería destacar la labor de mi tutor Pablo Castell que nuevamente ha estado conmigo el día a día, participando muy activamente en el trabajo, proponiendo nuevas ideas y lidiando con los problemas. Me siento muy afortunada de tener un guía que sepa tanto como él y se preocupe de sus tutorandos como él lo hace.

También quería dar las gracias a mis amigas y compañeras Julia y Cris que, después de aguantarme en el grado, decidieron unirse a la locura del máster también y compartieron conmigo numerosas tardes y fines de semana de prácticas interminables. No me quería olvidar de los chicos de la asociación de teleco, por acogerme en sus comidas. Muchas gracias Pencho, Manu, Erik, Rafa y todos los que hacéis que no haya un día en el que no salga riéndome de allí. También a Javi, Sofía y Nacho por los buenos ratos pasados en el laboratorio.

Por último, destacar la labor de mis padres y mi familia, porque siempre han estado ahí para darme su apoyo y comprensión. A ellos y a todas las personas que de una forma u otra han hecho posible la realización de este trabajo, lo sepan o no, muchas gracias.

Rocío

Glosario

Acierto: Tipo de métrica que evalúa el grado en que los recomendadores aciertan con los gustos del usuario.

Descubrimiento: Actividad lleva a cabo por los usuarios – voluntaria o involuntariamente – y que consiste en adquirir información acerca de la existencia de un ítem.

Popularidad: Término que se emplea en dos sentidos: como propiedad de los ítems que mide lo importantes – conocidos, relevantes, votados, etc. – que son, o como algoritmo que emplea dicha propiedad para recomendar aquellos en los que presenta un mayor valor.

Popularidad relevante: Noción de popularidad que sólo tiene en cuenta para cada ítem a los usuarios a los que gusta el ítem (por ejemplo, número de votos cuyo valor refleja una preferencia positiva).

Popularidad total: Noción de popularidad que no tiene en cuenta si a los usuarios les gusta o no el ítem en cuestión (por ejemplo, número total de votos, reflejen estos preferencia positiva o negativa indistintamente).

Precisión: Métrica de acierto aplicable a una recomendación para un usuario, o un conjunto de recomendaciones para un conjunto de usuarios, que cuantifica la tasa ítems relevantes recomendados (a un usuario individual, o promediada sobre un conjunto de usuarios).

Precisión observada: Precisión que se calcula considerando como relevantes los ítems presentes como tales en un conjunto de test para un usuario.

Precisión real: Precisión que se calcularía considerando como relevantes los ítems que gustan al usuario, con independencia de si el usuario ha manifestado dicho gusto o no. En la mayoría de los casos, la precisión real es una construcción teórica que no es viable obtener.

Ránking: Lista ordenada de recomendaciones.

Rating: Valoración numérica de los ítems realizada por los usuarios y que indica el nivel de relevancia que dichos ítems tienen para ellos. Pueden ser numéricos (en una determinada escala gradual) o binarios. También se emplea este término para referirse al acto de comunicar el usuario tal valoración al sistema.

Relevancia observada: Manifestación de los gustos de los usuarios – relevancia real – a través de las interacciones de dichos usuarios con el sistema. La relevancia observada representa, por tanto, una muestra de la relevancia real, la que el sistema conoce, observa.

Relevancia real (o relevancia a secas): Término que representa los gustos del usuario hacía los ítems. En este trabajo se considera, para simplificar, relevancia binaria, es decir, los ítems pueden gustar (ser relevantes) o no gustar (ser no relevantes) a los usuarios, sin distinción del posible grado de este interés.

Partición: División de los datos disponibles en dos conjuntos, uno de entrenamiento, que se da como entrada al recomendador, y otro de test, que se emplea para evaluar su capacidad para acertar con los gustos del usuario.

Índice

Índice de figuras

Índice de tablas

1. Introducción

1.1 Motivación

Desde sus inicios a principios de los años 90, los llamados sistemas de recomendación (Adomavicius & Tuzhilin 2005) han extendido progresivamente su presencia en tecnologías de uso diario, hasta ser hoy día un elemento familiar para los usuarios de aplicaciones, servicios y herramientas de los ámbitos más cotidianos. El común de los usuarios está acostumbrado hoy a que Youtube nos recomiende vídeos relacionados con nuestros intereses, que Spotify nos sugiera música que escuchar, que Twitter, LinkedIn o Facebook nos recomienden con quién conectarnos, que Google Play sugiera aplicaciones para nuestros smartphones, o que cualquier tienda online (Amazon, Fnac, etc.) recomiende compras que el sistema predice que nos pueden interesar en base a nuestros registros de compras y navegación.

La recomendación puede verse como la cara complementaria de los buscadores: en lugar de ser los usuarios quienes explícitamente expresan lo que les interesa con una consulta, es el sistema el que "busca" al usuario para emparejarle con aquello que – probablemente – satisface sus necesidades de información, tratando de determinar a quién le puede interesar una noticia, un producto, un servicio, un evento, una oferta de empleo, una conversación, o incluso otros usuarios. El sistema de recomendación toma la iniciativa, y dispone para sus predicciones de todo lo que sea capaz de observar en la actividad (compras, consultas, clicks, likes, etc.) del usuario que transcurre a través del sistema.

La popularidad es un concepto que aglutina diferentes perspectivas relevantes para los sistemas de recomendación. En el contexto que nos ocupa, popularidad en sentido coloquial significa aquello que gusta a mucha gente o que conoce mucha gente. Probar (por tanto recomendar) aquello que gusta a la mayoría parece cuando menos una idea razonable y útil en muchas ocasiones. La popularidad es un fenómeno social que puede resultar de muchos mecanismos de comportamiento e interacción humana, tales como la imitación y los procesos que se derivan de ella, como el aprendizaje social, la influencia social, la difusión de innovaciones, etc., en los que la adopción del comportamiento, la opinión o los descubrimientos de otras personas (ya sean semejantes, expertos o mayorías) nos resulta útil para beneficiarnos de la experiencia y lo aprendido por otros, guiarnos en situaciones de incertidumbre, o reducir el coste que implica elaborar una decisión desde cero (Miller & Dollard 1979). Este principio está en la base de la propia evolución de la especie humana, la civilización, y el desarrollo del individuo desde su primera infancia. Si bien muchas ramas de las ciencias sociales o la biología nos hacen entender lo amplias que pueden ser las diferencias entre dos individuos, que las tecnologías de personalización buscan precisamente tener en cuenta, no es menos cierto que desde una perspectiva amplia y objetiva es mucho lo que nos asemeja a un ser humano de otro, y no son pocas las ocasiones en las que lo que es bueno para uno lo es para el otro; dicho de otro modo, tenemos mucho en común con la mayoría de nuestros semejantes.

Por todo ello una recomendación basada en la popularidad de lo recomendado puede resultar aceptable o provechosa en muy diversas circunstancias. Desde la perspectiva de un especialista en sistemas de recomendación, recomendar un producto basado sólo en el número de personas que lo han consumido puede parecer una opción exageradamente simple y limitada. Sin

embargo son muy comunes los ejemplos donde se utiliza este mecanismo: prácticamente todas las plataformas web que ofrecen productos – Amazon, YouTube, periódicos, redes sociales – tienen un apartado dedicado a lo más popular – lo más visto, lo más leído, lo más comprado, etc. Más aún, la comunidad investigadora ha comprobado recientemente que la recomendación basada en popularidad resulta una opción inferior pero competitiva frente a algoritmos mucho más sofisticados (Cremonesi et al 2010), tanto en experimentos de laboratorio como en aplicaciones comerciales.

No solamente es la popularidad un criterio posible de recomendación por sí mismo. En el contexto de los sistemas de recomendación, la popularidad de un producto se traduce en el volumen de datos (derivados de la interacción de los consumidores) que el sistema posee sobre el mismo. A consecuencia de ello, en la práctica los algoritmos de recomendación (especialmente de filtrado colaborativo) basados en las observaciones de consumo tienden a sesgarse en mayor o menor medida hacia los productos que tienen suficientes datos como para posibilitar las predicciones del algoritmo, es decir, hacia las opciones populares (Marlin & Zemel 2009). La popularidad está pues presente en la práctica en gran medida, intencionadamente o no, en las tecnologías y aplicaciones de recomendación.

La popularidad presenta y plantea no obstante varias limitaciones y dudas. En primer lugar, recomendar lo popular, en sentido estricto, significa renunciar a la personalización e ignorar las particularidades de cada usuario, generando recomendaciones tal vez aceptables pero posiblemente subóptimas ("una talla para todos"). Por otra parte, el campo de las ciencias sociales ha estudiado desde hace más de un siglo (Trotter 1916) las derivas, ocasionales o frecuentes, de las mayorías hacia el error. Dicho de otro modo, es bien sabido que la mayoría no siempre acierta, cuando no yerra catastróficamente. Por citar un precedente más reciente y directamente relacionado con el contexto del consumo y la recomendación, un conocido estudio (Salganik et al 2006) comprobó cómo en condiciones de libre elección y sin influencias externas los mercados propenden hacia la concentración del consumo en determinados productos. Sin embargo se observó que el proceso está sujeto a un grado de incertidumbre por el cual, repitiendo el experimento sin variar las condiciones, el producto mayoritario no es siempre el mismo. El resultado parece depender de una combinación entre el gusto de los usuarios, la visibilidad del consumo total de cada producto, y un factor de azar en el orden de descubrimiento de los productos. Este estudio plantea así pues una duda legítima respecto a que el producto más consumido sea el que más gusta, pues ello implicaría que los gustos de los usuarios hubiesen variado durante la ejecución del experimento. La duda crece cuando consideramos mercados no exentos de influencias externas a los consumidores, como las campañas de márketing y otros factores habituales de sesgo en el acceso a los canales de comunicación y distribución, o los mecanismos de convergencia y conformidad social más allá de los gustos propios (Miller & Dollard 1979).

Así pues encontramos que está abierta en el campo de la recomendación la pregunta de hasta qué punto es la popularidad efectiva o un ingrediente deseable en un recomendador, en qué medida y bajo qué circunstancias. Si bien algunos autores han constatado el fenómeno de la aparente efectividad de la popularidad en la recomendación (Cremonesi 2010), e incluso han propuesto métricas y algoritmos que tienen en cuenta la popularidad (Steck 2010, 2011), no parece existir aún una comprensión clara de los efectos que la popularidad produce en la recomendación, ni una explicación de su efectividad o no como ingrediente de la misma.

El problema se complica teniendo en cuenta que el método principal para responder a estas preguntas, el experimental, se ve generalmente sujeto a los mismos sesgos que los propios datos que los recomendadores toman como entrada. La evaluación experimental proporciona una cota inferior del acierto de un recomendador, pues mide los aciertos observados, y no puede tener en cuenta los no observados (Herlocker et al 2004, Shani & Gunawardana 2011). Dado que las observaciones se distribuyen de modo desigual entre los productos, la subestimación del acierto es igualmente desigual entre productos, y recomendar un producto popular resulta más rentable, pues la probabilidad de que los aciertos a los que dé lugar sean observables es mayor a priori. El problema es más amplio que lo que respecta a la recomendación por popularidad pura, teniendo en cuenta que muchos algoritmos comunes de recomendación tienden a recomendar también productos populares.

Por último, cabe destacar que los ejemplos y casos anteriores son ventajas y limitaciones que atienden a la capacidad de acierto de la popularidad. En el sentido más amplio de la pregunta acerca de la conveniencia (efectividad) de recomendar opciones populares, cabría tener en cuenta otros aspectos más allá del puro acierto, como es muy en particular la falta de novedad o de diversidad. Sin embargo, en el presente trabajo nos vamos a centrar estrictamente en el aspecto del acierto, que consideramos que por sí mismo merece estudio y una mejor comprensión que la que actualmente existe.

En resumen, la investigación propuesta aborda las siguientes preguntas:

- ¿Es la popularidad realmente una señal eficaz para producir recomendaciones acertadas? ¿Es siempre mejor que una recomendación aleatoria, o podría llegar a ser peor?
- ¿De qué condiciones depende la respuesta a la pregunta anterior, y cómo es esta dependencia?
- ¿Puede llegar a discrepar la comparación entre la recomendación popular y aleatoria valorada en términos de relevancia observada y relevancia real?
- ¿Hay una diferencia significativa entre entender la popularidad como número total de votos o como únicamente los relevantes?
- ¿Puede variar la conclusión del experimento según cómo se realice el muestreo de datos de entrenamiento y test?

1.2 Objetivos

Partiendo del contexto y las preguntas anteriormente formuladas, el presente trabajo tiene como objetivo general el estudiar en qué medida y bajo qué circunstancias la recomendación por popularidad es una técnica eficaz o no. Para avanzar hacia dicho objetivo, el trabajo se divide en los siguientes objetivos específicos:

- Delinear un marco preciso en el que se definan los conceptos y distinciones sobre los que se asienta la recomendación por popularidad, la efectividad de la misma, y los diferentes posibles matices en estos conceptos.
- Caracterizar y cuantificar formalmente el acierto esperable de la recomendación por popularidad bajo distintas condiciones.

- Identificar, explicar y cuantificar los sesgos hacia la popularidad en la experimentación llamada *offline*.
- Identificar y explicar situaciones en las que el resultado de un experimento contradice el acierto real de la recomendación por popularidad.
- Relacionar el acierto de la popularidad con la distribución del descubrimiento de opciones por parte de los usuarios, y las posibles dependencias entre dicho descubrimiento y los gustos de los usuarios.
- Relacionar el acierto de la popularidad con el comportamiento del usuario a la hora de decidir interactuar con los productos, y las posibles dependencias entre dicho descubrimiento y los gustos de los usuarios.
- Relacionar el acierto de la popularidad con el diseño de la evaluación experimental offline, en particular con la separación de los datos en conjuntos de entrenamiento y test, y las posibles dependencias entre dicha separación y los gustos de los usuarios.
- Tener en cuenta explícitamente la interacción compleja entre el acierto obtenible en la recomendación y el descarte de productos ya consumidos (que no deben volver a recomendarse).
- Determinar, en definitiva, bajo qué condiciones la popularidad resulta o no una recomendación efectiva.
- Obtener conclusiones tanto teóricas como empíricas al respecto de todo lo anterior.

A fin de acotar el alcance del presente trabajo, no es parte de los objetivos del proyecto el paliar o corregir los sesgos de popularidad o sus efectos. El trabajo que aquí se presenta se enfoca a la formulación de los mismos, la comprobación de que existen, y la demostración de que pueden llegar a causar contradicción entre la efectividad medida y la real, objetivos que consideramos importantes de por sí. El diseño de medidas que compensen estos efectos se contempla pues, en todo caso, como trabajo futuro.

1.3 Estructura del trabajo

El documento se estructura de la siguiente forma:

- En el capítulo 2 (La popularidad en recomendación) realizamos una revisión empírica preliminar de la efectividad de la popularidad, comparándola con algoritmos representativos del estado del arte, y observando la existencia en dichos algoritmos de fuertes sesgos hacia lo popular.
- En el capítulo 3 (Estado del arte) exponemos una serie de conceptos relacionados con los sistemas de recomendación e introducimos los principales estudios que tratan acerca de la popularidad.
- En el capítulo 4 (Formulación teórica) deducimos una expresión analítica general para evaluar la precisión de un algoritmo de recomendación.

- En el capítulo 5 (Influencia de la distribución de popularidad) analizamos, a partir de la fórmula desarrollada en el capítulo anterior, la influencia de los sesgos de la distribución de popularidad en la evaluación de los algoritmos.
- En el capítulo 6 (Influencia del descubrimiento) estudiamos la influencia de la distribución de descubrimiento y el comportamiento del usuario en el resultado de la evaluación. Realizamos también un análisis empírico que engloba la formación de un conjunto de votos a partir de usuarios reales en unas ciertas condiciones de ausencia de sesgos, empleando para ello una plataforma de crowdsourcing.
- En el capítulo 7 (Influencia del protocolo de partición) analizamos la influencia de la forma en que se dividen los datos en entrenamiento y test.
- En el capítulo 8 (Ampliación) ampliamos, de forma empírica, el estudio realizado con los datos de la plataforma de crowdfsourcing a otros algoritmos y métricas.
- Para concluir, en el capítulo 9 (Conclusiones) resumimos el trabajo y sintetizamos las conclusiones que de él se extraen. También introducimos los posibles caminos a seguir en un trabajo futuro.

1.4 Notación

A lo largo del presente documento, especialmente en el desarrollo analítico, haremos uso de una cierta notación que – aunque explicaremos cada vez que emplee por primera vez – listamos a continuación para facilitar su consulta.

$\mathcal{U}$	Conjunto de todos los usuarios.
$\mathcal{I}$	Conjunto de todos los ítems.
m	Número de usuarios: $m = \lvert\mathcal{U}\rvert$.
n	Número de ítems: $n = \lvert\mathcal{I}\rvert$.
θ	Variable aleatoria que representa el recomendador.
$rpop$	Recomendador por popularidad relevante.
pop	Recomendador por popularidad total.
rnd	Recomendador aleatorio.
π	Variable aleatoria que representa el protocolo de partición.
δ	Variable aleatoria que representa la distribución de ratings.
$\mathcal{R}$	Variable aleatoria que denota el conjunto formado por las recomendaciones realizadas a todos los usuarios, entendiendo recomendación como una ordenación particular de todos los ítems.

R	Valor concreto de la variable aleatoria $\mathcal{R}$, por lo que representa un conjunto de recomendaciones concreto.
$\sigma(\mathcal{I})^m$	Conjunto de todas las posibles recomendaciones. Las permutaciones $\sigma(\mathcal{I})$ del conjunto de ítems "elevadas al número de usuarios", es decir **R** incluye una permutación de ítems – una recomendación, un ránking – $R \in \sigma(\mathcal{I})$ por usuario.
$\mathcal{S}$	Variable aleatoria que representa la partición (*split* en inglés).
s	Valor concreto de la variable aleatoria $\mathcal{S}$, por lo que denota una partición concreta.
$\mathbb{S}$	Conjunto de todas las posibles particiones.
u	Usuario concreto. En ocasiones se emplea para denotar el conjunto de ítems puntuados por (i.e. con los que ha interactuado) dicho usuario.
$\|u\|$	Número de votos introducidos por el usuario u (i.e. número de ítems que ha puntuado el usuario).
$\|u^{rel}\|$	Número de votos relevantes introducidos por el usuario u.
i	Ítem concreto. En ocasiones se emplea para denotar el conjunto de usuarios que han puntuado dicho ítem.
$\|i\|$	Número de votos recibidos por el ítem i (i.e. número de usuarios que han puntuado el ítem).
$\|i^{rel}\|$	Número de votos relevantes recibidos por el ítem i.
M	Métrica de evaluación. Cuando se refiere a la recomendación de un usuario concreto se emplea con el subíndice u: M_u.
P	Precisión real.
$\bar{P}$	Precisión observada.
R	Ránking de ítems, se emplea para denotar una recomendación concreta. Cuando se refiere a la recomendación de un usuario concreto se emplea con el subíndice u: R_u. Cuando queremos hacer referencia a las primeras k posiciones añadimos el superíndice k: R_u^k.
$\mathbb{I}_u$	Primer ítem recomendable. Es una función $\mathbb{I} = \tau(u, R_u, \pi, \delta)$ del usuario u, de su recomendación R_u, del protocolo de partición π, y de la distribución de ratings δ.
rel: $\mathcal{U} \times \mathcal{I} \to \{0,1\}$	Variable aleatoria que representa la relevancia, los gustos de los usuarios. Toma el valor 1 si el ítem i resulta relevante para el usuario u y 0 en otro caso.

$seen$: $\mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$	Variable aleatoria que representa el descubrimiento. Toma el valor 1 si el ítem i es conocido por el usuario u y 0 en otro caso.
$rate$: $\mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$	Variable aleatoria que representa la votación. Toma el valor 1 si el ítem i ha sido votado por el usuario u y 0 en otro caso.
$training$: $\mathcal{U} \times \mathcal{I} \times \mathcal{S} \rightarrow \{0,1\}$	Variable aleatoria que representa el concepto "estar en entrenamiento". Toma el valor 1 el usuario u ha votado el ítem i y el voto se encuentra en el conjunto de entrenamiento especificado por la partición s.
$test$: $\mathcal{U} \times \mathcal{I} \times \mathcal{S} \rightarrow \{0,1\}$	Variable aleatoria que representa el concepto "estar en test". Toma el valor 1 el usuario u ha votado el ítem i y el voto se encuentra en el conjunto de test especificado por la partición s.

2. La popularidad en recomendación

Antes de introducir los trabajos del estado del arte relevantes para este proyecto, y de cara a entender la trascendencia del mismo, empezamos por definir los conceptos y elementos básicos de la tarea de recomendación. Precisaremos a continuación las nociones necesarias para buscar respuestas a las preguntas de investigación que planteamos aquí y desarrollar nuestro estudio, en particular las definiciones y matices referidos a qué se entiende por popularidad y la efectividad de la misma. Pondremos así mismo en contexto la cuestión presentando unos resultados empíricos básicos que ilustren el sentido de nuestras preguntas y su importancia. En particular mostraremos una comparativa de la eficacia de la popularidad como método trivial de recomendación frente a otros algoritmos comunes del área, así como la tendencia de estos a sesgarse hacia aquélla.

2.1 La tarea de recomendación

Como hemos comentado en la introducción del presente trabajo, la recomendación puede verse como la cara complementaria de un motor de búsqueda, donde es el propio sistema el que toma la iniciativa para sugerir al usuario aquello que predice que puede satisfacerle. Las opciones susceptibles de ser recomendadas pueden ser de muy diverso tipo (noticias, servicios, productos, eventos, personas, etc.) por lo que se suele emplear el término genérico "ítems" para referirse a ellas.

Los sistemas de recomendación tratan por tanto de predecir los gustos de los usuarios, y en función de ellos anticipar qué ítem puede ser de su interés y recomendárselo. Para ello, el algoritmo de recomendación dispone de la manifestación de dichos gustos a través de las interacciones del usuario con el sistema. Pese a la gran variedad en la forma y tipo de dichas interacciones (la reproducción de una canción en Last.fm, la visualización de un video en YouTube, la asignación de una puntuación del 1 al 5 a una película en Netflix, la asignación de un "like" a un post en Facebook), es común representar de forma simplificada estos tipos de evidencia, sin gran pérdida de generalidad a muchos efectos, como la asignación de un voto explícito al ítem por parte del usuario, ya sea binario (me gusta / no me gusta) o numérico (en una escala gradual de menor a mayor gusto por el ítem), al que nos referiremos indistintamente a lo largo del documento como rating, voto, puntuación o interacción. Cuando el voto refleja una preferencia favorable al ítem diremos que el ítem es *relevante* para el usuario. En este trabajo adoptaremos esta representación simplificada en forma de ratings, y tomaremos la simplificación adicional de que los votos sean binarios (relevante / no relevante), pues es suficiente para nuestros fines.

Dado un escenario como el descrito en el que el sistema dispone de un conjunto de ratings asignados por los usuarios a los ítems, la tarea de recomendación consiste en seleccionar los ítems que un usuario no ha votado y ordenarlos en una lista (ránking) de forma descendiente según la relevancia que el algoritmo estima que cada ítem puede tener para el usuario. En la definición anterior estamos considerando que no se permiten recomendaciones repetidas, es decir, no se recomienda aquello que el usuario ya ha votado. Este suele ser el planteamiento usual, siendo muy pocos aquellos que consideran situaciones en las que es posible recomendar

ítems que los usuarios ya han consumido (Benson et al 2016). El motivo de nuestra asunción es que en la mayoría de dominios no se suele repetir la consumición de un mismo ítem, por lo que si el usuario ya lo ha consumido no tiene sentido volver a recomendarlo. Existen sin embargo casos en los que sí se puede consumir un mismo ítem varias veces – por ejemplo, se puede escuchar varias veces la misma canción en Spotify – pero en dichos casos suponemos que el recomendador se enfoca en la tarea de descubrir nuevos ítems, por lo que los ya conocidos no se recomiendan.

La recomendación tiene pues como objetivo principal, al menos en teoría, maximizar la satisfacción de los destinatarios de las recomendaciones. Naturalmente este objetivo debe conjugarse en la práctica con otras prioridades como los costes del algoritmo y el rendimiento del negocio de quien financia el desarrollo y despliegue de las funcionalidades de recomendación en una aplicación concreta, pero nos abstraeremos aquí de tales aspectos que se desvían de nuestro objetivo. No es posible en cualquier caso considerar de forma aislada el desarrollo de los algoritmos de recomendación de la forma en la que va a ser evaluada su efectividad, pues la selección del algoritmo óptimo puede variar ampliamente según el método de evaluación que se aplique. En el presente estudio se van a considerar las metodologías de evaluación offline de uso común actualmente en el campo de los sistemas de recomendación. Estas metodologías dividen los datos disponibles en un conjunto de entrenamiento, que se da como entrada a los algoritmos de recomendación, y un conjunto de test, que se oculta a los algoritmos y se utiliza para simular la evidencia de gustos de los usuarios no observados por el algoritmo de recomendación, y comprobar la capacidad de los algoritmos para acertar con ellos, utilizando métricas de acierto en el ránking como precisión, recall, nDCG, etc. (Baeza & Ribeiro 2011). El empleo de parte de los datos manifestados por los usuarios para evaluar la capacidad de acierto de los recomendadores supone considerar como fallos aquellas opiniones que no se conocen (pudiendo ser estas realmente positivas), lo cual motiva la distinción entre efectividad medida y real.

Es amplia y conocida la cantidad de algoritmos que se han desarrollado desde hace más de dos décadas para resolver el problema de la recomendación. Comentaremos muy brevemente las principales estrategias en la sección 3.1. Frente a los métodos sofisticados que buscan proporcionar la mejor recomendación posible a la medida de cada usuario, existen alternativas más sencillas aunque no por ello menos utilizadas. Una muy común es la que nos va a ocupar en este trabajo, la recomendación por orden inverso de popularidad. Antes de analizar la efectividad de la popularidad como criterio de recomendación, debemos precisar claramente las posibles definiciones de este criterio, como hacemos a continuación.

2.2 ¿Qué se entiende por popularidad?

La noción de popularidad admite diferentes interpretaciones con ligeras diferencias de matiz, tal como puede uno comprobar si busca el término en cualquier diccionario. Por lo general la popularidad de un ítem hace referencia a la percepción global que el ítem tiene por parte de una población de personas (o entidades), donde los elementos definitorios de dicha percepción son el volumen (número de usuarios) y el signo (grado de positividad) de dicha percepción. En el contexto de la recomendación, la consideración de uno, otro o ambos elementos da lugar a tres posibles definiciones principales de la popularidad de un ítem, que denominaremos como *popularidad total*, *popularidad relevante*, y *voto promedio*.

Estas nociones de popularidad se definen en base a dos relaciones elementales entre usuarios e ítems: el gusto por el ítem, y la expresión de este gusto en el sistema en forma de rating. Cabe considerar una noción más: el conocimiento del ítem por parte del usuario, independientemente de que el sistema haya observado interacción entre ambos. Esta relación es difícil de obtener en general de modo completo, y por ello se toma comúnmente la relación de rating como aproximación o (sub)estimación de lo que los usuarios conocen: los ratings son la muestra observada por el sistema de los ítems que los usuarios conocen. A efectos de los objetivos del presente trabajo no precisaremos hacer esta distinción entre conocimiento (descubrimiento) y rating en la definición de popularidad.

A continuación se explica con más detalle cada una de estas interpretaciones, así como las posibilidades en las que derivan a la hora de tomar la popularidad como criterio de recomendación.

2.2.1 Popularidad como cantidad de votos

Entender la popularidad de un ítem como el número de votos recibidos por el mismo es la interpretación más sencilla y frecuente de la popularidad en la literatura de los sistemas de recomendación (Cremonesi et al 2010). Es por ello la que se va a analizar en este trabajo, dejando el estudio del rating promedio como trabajo futuro.

Como ya adelantábamos, la distinción entre votos relevantes y no relevantes permite a su vez considerar dos tipos de popularidad: popularidad total y popularidad relevante.

Popularidad total

La popularidad se define como el número total de ratings de un ítem, independientemente de si estos son positivos o negativos. Esta es la definición usual de popularidad que se suele encontrar en la literatura. Pese a que ignorar el signo de los votos podría parecer a priori una simplificación importante, en los experimentos documentados en la literatura no parece dar lugar a una diferencia perceptible a la hora de utilizar la popularidad como criterio de recomendación, posiblemente porque el número de votos negativos es habitualmente bajo en los conjuntos de datos típicos, y porque suele existir además una tendencia positiva entre el número de votos y la proporción de votos favorables (Pradel et al 2012). No obstante, en un entorno en que se votase más a menudo lo que no gusta que lo que gusta, la popularidad total podría acabar recomendando los ítems que menos satisfacen a los usuarios.

Popularidad relevante

La popularidad relevante se define como el número de votos cuyo valor representa una preferencia positiva, es decir, el número de usuarios a quienes gusta el ítem. Como acabamos de mencionar, pese a que esta definición parece más lógica que la popularidad total como criterio de recomendación, en la mayoría de conjuntos de datos utilizados en el estado del arte se ha visto que no hay diferencias significativas entre recomendar de una forma o de otra. Sin embargo, en otros tipos de situaciones ambos modelos de popularidad pueden presentar comportamientos sensiblemente distantes entre sí, cómo se verá a lo largo de este trabajo.

2.2.2 Popularidad como rating promedio

Es muy frecuente que aplicaciones y portales Web en los que se recaba la opinión de los usuarios sobre un catálogo de productos (Amazon, Google Play, IMDb, Guía del Ocio, etc.) muestren la valoración de dichos productos como el promedio de los votos recibidos (por ejemplo en una escala de 1 a 5 estrellas). En estos casos se mide lo popular que es un ítem en

términos del rating promedio. En el caso de la relevancia binaria, el rating promedio se reduce a la tasa entre el número de votos relevantes y el número total de votos recibidos.

Pese a lo frecuente que es emplear esta medida para informar al usuario acerca de lo popular que es un ítem, a la hora de recomendar suele presentar una efectividad bastante baja en comparación con la interpretación de la popularidad cómo número total de votos. El motivo es que presenta un fuerte sesgo hacia los ítems poco populares cuyos pocos votos sean todos relevantes. Aunque los ítems impopulares no tengan muchos votos positivos, es mucho más fácil obtener la máxima puntuación promedio con pocos que con muchos votos. Por ejemplo, si un ítem únicamente ha recibido un voto y este es relevante, su popularidad ya es máxima según el rating promedio. Por el contrario, es muy difícil obtener la máxima puntuación con un promedio de muchos votos, pues tendrían que ser unánimes y esto es mucho más difícil de conseguir. Como resultado los, pongamos, diez primeros ítems del ránking por rating promedio tienen un fuerte sesgo hacia ser los ítems con menos votos, que no sin embargo resultan ser muy efectivos para la evaluación de la recomendación. Con el objetivo de evitar este efecto, generalmente se suele establecer un número mínimo de votos para poder recomendar un ítem, es decir, es necesario hibridarlo con popularidad en el sentido del número de votos.

De aquí en adelante emplearemos el término "popularidad" para referirnos a la primera interpretación de popularidad como cantidad de votos, dejando el término "rating promedio" para designar la segunda interpretación.

2.3 Efectividad de la popularidad

Como ya se introdujo en la sección 2.1, la recomendación por popularidad es una recomendación no personalizada con una algoritmia bastante simple: recomendar siempre los ítems más populares. Pese a esta simplicidad, llama la atención la alta efectividad que suele presentar en la literatura cuando se evalúa empleando las metodologías de evaluación offline basadas en ránking (Cremonesi et al 2010).

Para poner en contexto la investigación que aquí se presenta, empezamos por mostrar unos experimentos básicos sobre conjuntos de datos públicos de uso común en el área, donde medimos y comparamos la efectividad de la recomendación por popularidad con la de otros algoritmos de recomendación representativos del estado del arte. En concreto, vamos a emplear los conjuntos de MovieLens, Netflix y Last.fm, que se encuentran entre los más conocidos y utilizados en los estudios de recomendación.

MovieLens[1] es un conjunto de datos públicos de puntuaciones por usuarios a películas. Se trata posiblemente del dataset más utilizado en la literatura y la tradición investigadora en el campo de los sistemas de recomendación. En el caso de Netflix[2], se trata de datos suministrados por la conocida empresa en 2006 en el contexto del premio Netflix[3], en el que los ítems sobre los que se realizan las votaciones son películas y series de televisión. Finalmente utilizamos datos de Last.fm[4] elaborados por O. Celma (Celma 2010), donde los ítems son canciones. En la Tabla 1 mostramos los datos volumétricos de estos tres conjuntos de datos.

[1] http://grouplens.org/datasets/movielens
[2] file://raptor.ii.uam.es/collections/datasets/Datasets/netflix/others/viewtopic.php.htm
[3] http://www.netflixprize.com/
[4] file://raptor.ii.uam.es/collections/datasets/Datasets/Last.fm/360K%20Users/index.html

Dataset	Nº usuarios	Nº ítems	Nº ratings
MovieLens	6.040	3.706	1.000.209
Netflix	480.189	17.770	100.480.507
Last.fm	992	176.892	904.309

Tabla 1. Dimensiones de MovieLens, Netflix y Last.fm.

Como es común en los escenarios típicos en los que operan los sistemas de recomendación, y al igual que en todos los conjuntos de datos públicos conocidos, la distribución del número de votos por ítem presenta fuertes sesgos en los tres conjuntos de datos que utilizamos aquí. Esto se aprecia claramente en la Figura 1, que muestra la distribución de ratings de MovieLens indicando, para cada ítem (eje *x* ordenado por popularidad total decreciente), el número de usuarios que lo han votado (*eje y*). Observamos la existencia de unos pocos ítems votados por muchos usuarios mientras que la mayoría de ítems reciben muy pocas valoraciones. Este tipo de distribución en la que unos pocos productos populares concentran la atención de los usuarios, mientras que el resto de productos forman una larga cola de elementos poco o menos conocidos se denomina distribución pseudo-Pareto.

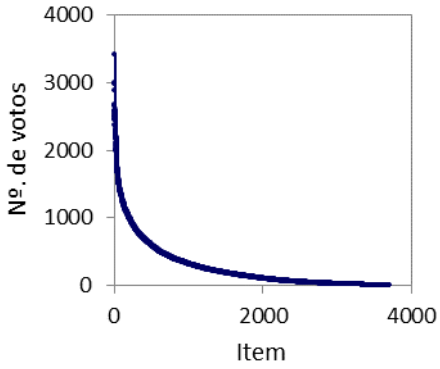

Figura 1. Distribución de ratings de MovieLens (en escala lineal).

Cuando el sesgo de la distribución se extrema, empieza a visualizarse mal en escala lineal y resulta más informativa la escala logarítmica, donde las distribuciones pseudo-Pareto se aproximan a un comportamiento lineal. En la Figura 2 se muestran las distribuciones de MovieLens, Netflix y Last.fm en dicha escala. Observamos que las dos últimas presentan una forma similar a MovieLens, pero con mayor sesgo, pues se asemejan más – especialmente Last.fm – a una recta.

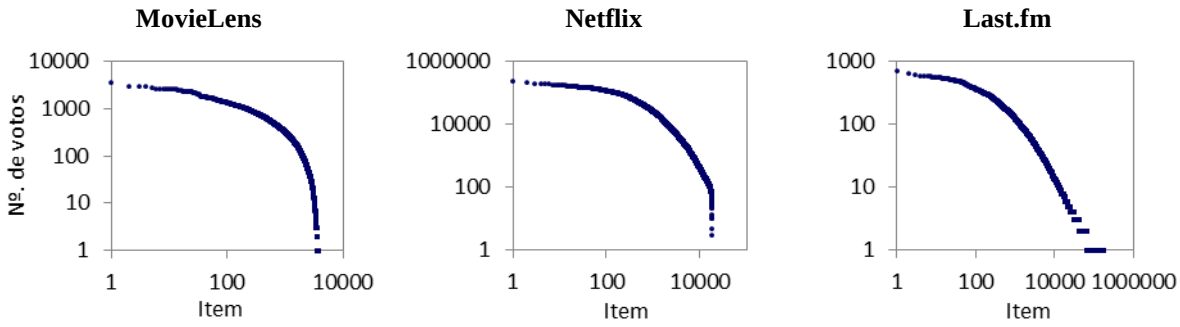

Figura 2. Distribución de ratings de MovieLens, Netflix y Last.fm (en escala logarítmica).

Para nuestra comparativa con popularidad seleccionamos dos métodos clásicos de recomendación por filtrado colaborativo: vecinos próximos (kNN, *k nearest neighbors*) y factorización de matrices (Adomavicius & Tuzhilin 2005). Junto con ello, utilizamos dos algoritmos más como punto de referencia: rating promedio (versión simple, sin requerir un número mínimo de ratings), y recomendación aleatoria.

Los algoritmos kNN de vecinos próximos presentan numerosas variantes que dependen del tipo de similitud, del número de vecinos que se toman, si se aplica o no normalización en la función de estimación de ratings, si están basados en usuario o en ítem, etc. (Ning et al 2015). Para nuestra comparativa consideramos los kNN basados en ítem (ib) y en usuario (ub), con y sin normalización, es decir cuatro algoritmos en total. Tras una somera exploración para seleccionar una configuración óptima en acierto, fijamos el número de vecinos en 50 y como similitud empleamos Jaccard (Amatriain & Pujol 2015).

Respecto al algoritmo de factorización de matrices, utilizamos el algoritmo propuesto por Hu et al. (2008), el más efectivo entre los probados en los últimos años por el grupo de investigación en el que se ha realizado el presente trabajo, y uno de los más rápidos en tiempo de ejecución. Como configuración de los parámetros del algoritmo tomamos 50 factores, $\lambda = 0.1$, $\alpha = 1.0$ y 20 iteraciones. Esta parametrización se ha tomado teniendo en cuenta experimentos previos del grupo en los que la exploración de parámetros ha mostrado un buen funcionamiento con estos valores.

Para paliar los efectos de la varianza de la partición realizamos una validación cruzada sobre 5 particiones. Como métrica de acierto, escogemos simplemente $P@10$. No precisamos más métricas en este punto, pues nuestra intención es simplemente presentar una simple panorámica del comportamiento típico de los algoritmos, y no una comparativa minuciosa a fondo de qué algoritmo es mejor o en cuántos casos.

En la Figura 3 se muestra la precisión en los diez primeros ítems recomendados ($P@10$) de los algoritmos anteriores y los dos tipos de popularidad – total y relevante – para los tres conjuntos de datos – MovieLens, Netflix y Last.fm. Los resultados confirman que la popularidad presenta una efectividad relativamente elevada, teniendo en cuenta su simplicidad. Es mucho mayor que los otros dos recomendadores no personalizados de referencia (aleatorio y rating promedio) y aunque es claramente subóptima respecto a los algoritmos de filtrado colaborativo (kNN no normalizados y la factorización de matrices), resulta llamativo que la diferencia no es aplastante. Se comprueba así mismo que la popularidad total y relevante presentan un acierto muy parecido, con una muy ligera ventaja de esta última.

Llama la atención también la baja precisión obtenida por el rating promedio, pues es inferior incluso al recomendador aleatorio. Esto se debe a su tendencia a recomendar lo menos popular cuando no se hibrida con popularidad, como ya hemos comentado en la sección 2.2.2. Los algoritmos kNN normalizados sufren de este mismo fenómeno pues la normalización los aproxima al rating promedio y por ello presentan un rendimiento igualmente pésimo. kNN sin normalizar es la segunda opción más efectiva después de la factorización de matrices. La versión normalizada se ha utilizado históricamente porque resulta efectiva cuando la recomendación se evalúa en términos del error de predicción del valor de rating (con métricas como MAE y RMSE). Sin embargo en los últimos años se está entendiendo que la predicción de rating por sí misma (y por lo tanto las métricas que la evalúan) no es un objetivo relevante en tanto que no correlaciona con la satisfacción del usuario en aplicaciones reales de

recomendación., y lo es más la calidad del ránking (que miden métricas como precisión), en cuyos términos comprobamos que estos algoritmos normalizados resultan muy subóptimos.

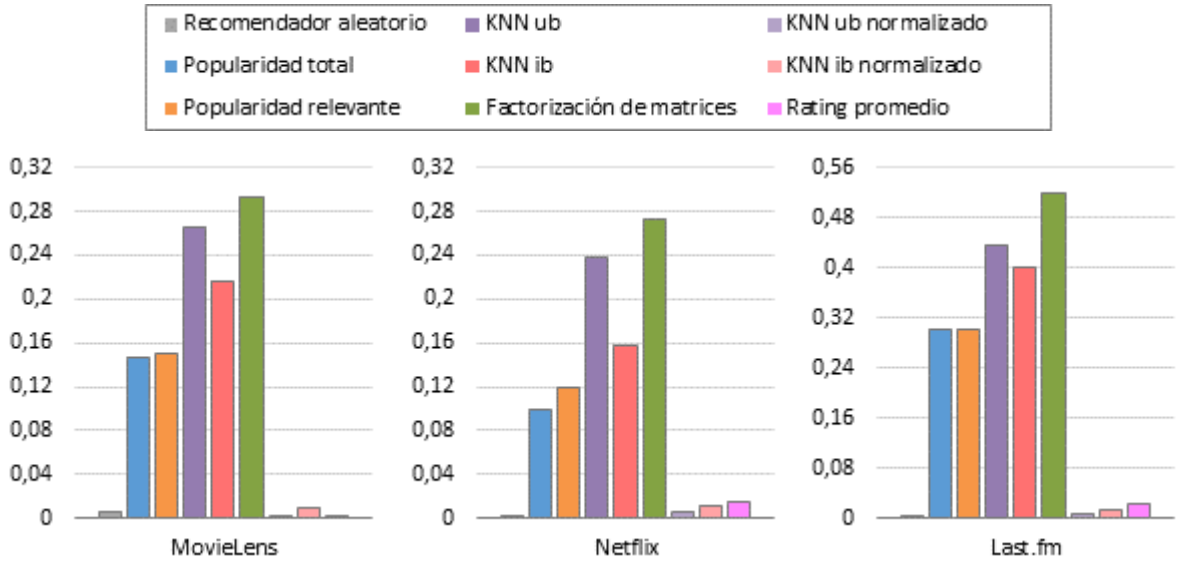

Figura 3. *P*@10 de varios recomendadores al ser evaluados sobre los conjuntos de MovieLens, Netflix y Last.fm.

2.4 Sesgo de los algoritmos hacia la popularidad

En la sección anterior hemos constatado la efectividad que puede conseguirse recomendando simplemente ítems populares, efectividad que no se aleja en exceso de la que se obtiene con algoritmos personalizados más elaborados, de los que cabría esperar una ventaja mucho mayor.

Este fenómeno ya resulta interesante de por sí, pero no es el único que justifica nuestro interés por la popularidad en los sistemas de recomendación. Se ha observado en estudios previos (Marlin & Zemel 2009) que existe un sesgo en los algoritmos de recomendación hacia los ítems populares. Comprobemos pues brevemente, en estos experimentos preliminares, en qué medida existe una componente de popularidad en los algoritmos personalizados, o dicho de otro modo, en qué medida se observa algún parecido entre las recomendaciones personalizadas y la recomendación por popularidad. En concreto, y dado que ambos tipos de popularidad – total y relevante – presentan en los conjuntos anteriores una precisión muy similar, la comparativa se realizará considerando la popularidad relevante de los ítems.

En la Figura 4 se visualiza el sesgo a lo popular que presentan los algoritmos mostrados en la sección anterior. El sesgo se visualiza como el número de veces que un algoritmo recomienda cada ítem en el top 10 del ránking (eje *y*) frente a la popularidad del mismo (eje *x*), para los tres conjuntos de datos (MovieLens, Netflix y Last.fm).

Observamos que kNN y (más aún) la factorización de matrices presentan un fuerte sesgo hacia la recomendación de ítems populares. El sesgo es más fuerte en kNN ub que ib. Vemos por otra parte que los ítems más recomendados por rating promedio presentan muy pocos votos relevantes, pero precisamente por ser pocos, su rating promedio obtiene un valor muy alto desvirtuando la recomendación. Observamos que los kNN normalizados siguen esta misma tendencia contraria a popularidad, lo cual concuerda con la baja precisión que obtienen, al mismo nivel que la del rating promedio. Vemos como curiosamente los algoritmos menos

efectivos son los que se desconectan de la popularidad, y los más efectivos parecerían utilizar indirectamente la popularidad, en cierta medida, como parte de su operativa.

Entender por tanto los factores de los que depende la efectividad de la popularidad no es únicamente relevante para el empleo de dicho algoritmo, sino que permite también comprender el comportamiento de otros muchos recomendadores más complejos. Dicho de otro modo, puesto que los algoritmos de filtrado colaborativo se parecen a la popularidad, o la tienen como una componente implícita en el algoritmo, las propiedades de esta pueden explicar propiedades del comportamiento de aquellos, y cualquier hallazgo que podamos realizar sobre la popularidad podría ser aplicable a otros algoritmos personalizados no triviales. Por ejemplo, si descubriésemos que la aparente efectividad de la popularidad no es tal (porque se debiera, por ejemplo, a una distorsión de la metodología), la misma duda recaería sobre otros algoritmos menos triviales, que propenden a la popularidad.

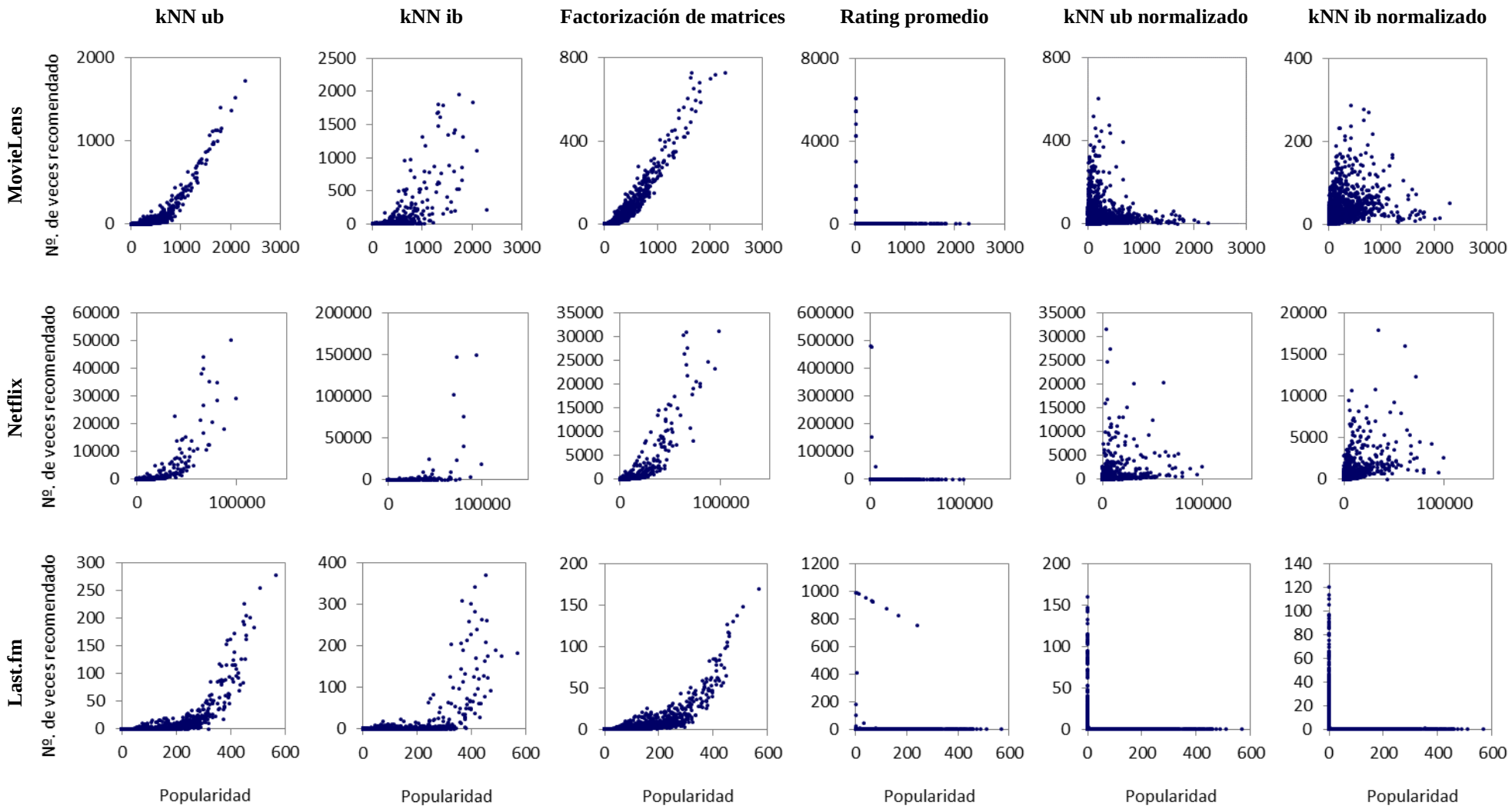

Figura 4. Número de veces que se recomienda cada ítem frente al número de votos relevantes (popularidad relevante) que presenta dicho ítem, para diversos recomendadores y para los conjuntos de MovieLens, Netflix y Last.fm.

3. Estado del arte

Introducimos y contextualizamos aquí una serie de conceptos necesarios para entender la situación del problema que se trata en el presente trabajo. En concreto, hablamos de los diferentes tipos de algoritmos de recomendación y de la forma de evaluarlos, aspectos relevantes en la medida en que el objetivo de este trabajo es analizar la efectividad de un algoritmo de recomendación concreto. A continuación nos enfocamos en la popularidad, entendiéndola como la distribución del número de votos de los ítems y como método de recomendación.

3.1 Algoritmos de recomendación

Los sistemas de recomendación se han consolidado a lo largo de las dos últimas décadas como área de investigación diferenciada (Ricci et al 2015), acompañada por un fuerte impacto y desarrollo comercial (Linden et al 2003). El desarrollo algorítmico ha sido muy amplio en este campo, por lo que resulta inviable y poco relevante para este trabajo mencionar las características de cada recomendador por separado. Sí es pertinente, sin embargo, proporcionar un marco en el que encuadrar los distintos algoritmos que se mencionan a lo largo del documento.

Así pues, en función de los datos de entrada del recomendador y su forma de interpretarlos, podemos clasificar los recomendadores en dos grandes tipos:

- Personalizados: tienen en cuenta las características individuales de cada usuario, además de otros factores, para adivinar sus intereses futuros. Dentro de los recomendadores personalizados existen varios subgrupos:
  - Recomendadores de filtrado colaborativo: en base a las opiniones del resto de usuarios y las similitudes que presentan con las valoraciones del usuario, se adivina su opinión acerca de los ítems que no conoce pero que personas con opiniones similares han valorado. Los algoritmos de vecinos próximos y factorización de matrices citados en el capítulo anterior se encuadran dentro de este grupo.
  - Recomendadores basados en contenido: se basan en las características de los ítems valorados por el usuario para estimar su opinión acerca de otros con características similares. No tienen en cuenta las opiniones de otros usuarios.
  - Recomendadores basados en red social: se basan en la opinión de los contactos explícitos del usuario en la red social para predecir los intereses del usuario.
  - Recomendadores híbridos: son combinaciones de los anteriores que equilibran sus virtudes y debilidades.
- No personalizados: recomiendan lo mismo a todos los usuarios, sin tener en cuenta sus características individuales. Estas recomendaciones pueden realizarse en base a muchos factores, entre los cuales destacan la opinión de la mayoría, las críticas de los expertos o las prioridades del proveedor que tiene interés en promocionar un cierto producto. La recomendación por popularidad o el rating promedio se encuentran en este grupo.

3.2 Evaluación

Hay múltiples formas de medir la calidad de la recomendación generada por un algoritmo, dependiendo de lo que se busque optimizar (Herlocker et al 2004, Shani & Gunawardana 2015). A continuación se exponen las principales.

Las métricas de error han sido durante mucho tiempo, y por herencia histórica del campo de la minería de datos, las más utilizadas para medir la eficacia de un sistema de recomendación. Se centran en el valor de las puntuaciones que el recomendador predice y miden qué tan cerca están respecto a la puntuación real que el usuario ha dado al ítem objetivo. Recientemente se está comprendiendo (Cremonesi et al 2010, Steck 2013,) que estas métricas no son representativas de la satisfacción real de los usuarios con las recomendaciones (dicho de otro modo, no correlacionan bien con la probabilidad de que los usuarios las acepten).

La industria basa actualmente sus evaluaciones principalmente en metodologías de test A/B en vivo sobre sus propias plataformas de producción, en las que la efectividad de la recomendación se mide fundamentalmente en términos del incremento de respuesta (compra, clickthrough, engagement) por parte de los usuarios sobre los elementos recomendados, la fidelidad de los clientes, y el rendimiento económico para el negocio. La comunidad académica por su parte, a falta de la capacidad para realizar tales pruebas, lleva a cabo sus experimentos sobre datos de prueba offline, adoptando recientemente metodologías y métricas propias del campo de la Recuperación de Información, tales como la precisión, recall o nDCG (Baeza & Ribeiro 2011), que miden la calidad del ránking de recomendación y parecen representar mejor que el error de predicción la utilidad final para los consumidores. Con estas métricas, además, se abre la puerta a la evaluación de algoritmos basados en ránking – como la popularidad – que no podían ser medidos en términos del error en la predicción del rating.

Como ya introducimos en la sección 2.2 anterior, las metodologías de evaluación offline (tanto las orientadas a error como a ránking) dividen los datos disponibles en un conjunto de entrenamiento, que se da como entrada a los algoritmos de recomendación, y un conjunto de test, que se emplea para comprobar la capacidad de los algoritmos para acertar con los gustos del usuario. En fechas recientes se viene observando cómo la forma de llevar a cabo esta separación (en definitiva, un muestreo) entre entrenamiento y test puede condicionar el resultado de la evaluación de algoritmos (Bellogín et al 2011). Más aun, la propia recogida de datos de interacción de los usuarios que tiene lugar al crear un conjunto de datos es en sí misma en definitiva un muestreo, y como tal se ve generalmente sujeta a sesgos cuya naturaleza y explicación no han sido prácticamente investigadas hasta la fecha.

De hecho, la literatura de los sistemas de recomendación apenas ha cuestionado el origen y las características de los datos que los algoritmos toman como entrada, que típicamente se toma como estándar de evaluación y comparación no cuestionado. Se ha considerado por supuesto como objeto de estudio la fiabilidad y la calidad de los datos (Cheng & Hurley 2009), pero no así su distribución y sesgos. No obstante, sí existen trabajados enfocados a constatar la existencia de dichos sesgos tal como veremos en detalle en la siguiente sección.

3.3 La popularidad en los sistemas de recomendación

Los trabajos realizados relacionados con las cuestiones de popularidad que aquí nos ocupan pueden agruparse en cuatro áreas generales, en función del aspecto que abordan: constatar la existencia e influencia de los sesgos de popularidad en la recomendación, proponer métricas que los tengan en cuenta y que palien sus efectos, reproducir el proceso que los genera o tratar el tema de la falta de novedad en la recomendación por popularidad. A continuación resumimos los principales trabajos pertenecientes a cada grupo. Tras ello resumiremos un estudio previo realizado por la autora del presente trabajo y su tutor, relacionado con el actual trabajo.

3.3.1 Existencia e influencia de los sesgos de popularidad

Como hemos mencionado en la sección anterior, la incorporación y utilización de las métricas de ránking – precisión, recall, nDCG etc. – ha permitido evaluar la efectividad de algoritmos que, como la popularidad, no habían podido ser evaluados empleando únicamente las métricas de error. Un estudio pionero en este sentido (Cremonesi et al 2010) comparó ambas formas de medir y constató que empleando las métricas de ránking, la recomendación por popularidad obtenía un rendimiento llamativamente elevado comparado con el de otros algoritmos más complejos y personalizados. El estudio de Cremonesi et al mostró además que este rendimiento disminuye cuando se eliminan los ítems más populares, de lo que puede deducirse que la efectividad de la popularidad tiene relación con el sesgo de la distribución de la misma (Cremonesi et al 2014). Sin embargo Cremonesi et al no desarrollan un análisis ni dan una explicación de esta relación.

Dichos sesgos se observan no solamente en la distribución desigual de votos, sino también en el hecho de que los votos positivos suelen ser más frecuentes que los negativos, es decir, la ausencia de ratings no es uniforme, depende del valor del mismo (Marlin et al 2007, Steck 2013, Pradel et al 2012, Goel et al 2010). Marlin et al (2007) fueron los primeros en constatar este fenómeno al preguntar explícitamente acerca de ello a usuarios reales. Así pues, una gran mayoría de dichos usuarios manifestaron que su opinión sobre los ítems afecta considerablemente a la decisión acerca de votarlos.

La desigual distribución de los datos no influye únicamente en la recomendación por popularidad, sino que tiene también un llamativo impacto en otros algoritmos como los métodos colaborativos, que tienden a recomendar ítems populares. Nuestras observaciones de la sección 2.4 en este sentido concuerdan con las de otros autores que ya observaron este efecto (Cremonesi et al 2010, Marlin & Zemel 2009).

Los estudios anteriores muestran por tanto que la ausencia de votos no se distribuye uniformemente y que ello influye sustancialmente tanto en las recomendaciones (se sesgan hacia lo popular) como en el resultado de la evaluación cuando se emplean métricas de ránking (se premia recomendar lo popular). Sin embargo, no se ha explorado aún la caracterización y explicación de las causas que dan lugar a diferentes sesgos, ni hasta qué punto tales sesgos provocan una distorsión en los resultados de un experimento (no sólo en el valor de las métricas sino también en particular en cuanto al signo de la comparación entre algoritmos), o por el contrario preservan las conclusiones del mismo. En el presente trabajo estudiamos precisamente las variables de las que dependen dichos sesgos y presentamos distintas situaciones – caracterizadas por la relación entre dichas variables – en las que se puede

comprobar la existencia o no de contradicciones entre la eficacia medida a partir de los datos observados y la que se tiene realmente.

3.3.2 Métricas que tienen en cuenta los sesgos de popularidad

Partiendo de constataciones como las que resumimos en la sección anterior, trabajos recientes han propuesto mecanismos de compensación en los algoritmos y métricas que tengan en cuenta la desigual distribución de datos sobre el conjunto de opciones a recomendar (Steck 2010, Steck 2011, Zhao et al 2013). Estos estudios abordan la cuestión de los sesgos de popularidad y en particular su influencia en los métodos de filtrado colaborativo como un problema que se debe resolver, argumentando que una recomendación que incluya ítems menos populares resulta más valiosa.

En particular, Zhao et al (2013) consideran que votar algo que a poca gente le gusta aporta más información acerca de los gustos del usuario que un voto a un ítem popular. Consecuentemente, a la hora de recomendar realizan ponderaciones que incrementan el peso de los votos a los ítems menos populares. En la misma línea, Steck (2010) define funciones objetivo que premian la recomendación de los ítems menos populares y que emplea tanto para evaluar como para entrenar los algoritmos. Sin embargo, el propio Steck (2011) advierte un año más tarde acerca de sesgar la recomendación hacia ítems poco populares, pues únicamente un pequeño sesgo en este sentido es apreciado por los usuarios.

Es relevante también para nuestro estudio el procedimiento que sigue Steck (2010, 2011) para definir las métricas y funciones objetivo. Steck se plantea medir o aproximar una estimación de la eficacia real de las recomendaciones, la que tuviese en cuenta los gustos reales y completos de los usuarios y no únicamente los observados. Sin embargo, ante la imposibilidad de utilizar los datos que no se conocen, propone una serie de métricas en base a una asunción principal: los votos relevantes sí se distribuyen uniformemente entre los ítems. Esta asunción le permite afirmar que las métricas que propone, al evaluarse sobre los datos observados, estiman correctamente los valores reales.

3.3.3 La falta de novedad en la recomendación por popularidad

Una de las principales limitaciones que plantea la recomendación de los ítems más populares es la falta de novedad (Oh et al 2011, Lee & Lee 2011, Celma & Herrera 2008, Nakatsuji et al 2010, Onuma et al 2009). Dichos ítems tan populares son generalmente conocidos por los usuarios, aunque no los hayan votado, por lo que recomendarlos carece de utilidad.

Existen numerosos trabajos enfocados en paliar esta deficiencia, como los citados en el párrafo anterior. En nuestro caso, sin embargo, antes de descartar la popularidad por su falta de novedad creemos que es relevante entender qué ocurre con su nivel de acierto, entre otras razones porque como venimos comentando presenta una notable influencia en otros algoritmos que obtienen generalmente un acierto muy elevado. Por ello, en el presente trabajo nos vamos a centrar en estudiar esa capacidad de acierto de la recomendación por popularidad, tema ya de por sí bastante complejo.

3.3.4 Proceso de generación de la popularidad

La distribución de popularidad de los ítems se deriva del proceso de generación de ratings, que lleva a dichos ítems a acumular un cierto número de votos. Aunque este proceso ha sido estudiado con anterioridad, son trabajos que se centran principalmente en intentar predecir y modelizar formalmente el comportamiento de los usuarios (Harper et al 2005, Borghol et al

2011), o en predecir la popularidad alcanzada por los ítems (Szabo & Huberman 2010, Shen et al 2014, Zhang et al 2014, Ratkiewicz et al 2010, Hensinger et al 2013), y no tanto en comprender cómo se derivan de dicho proceso los sesgos de la distribución de votos ni si dichos sesgos pueden distorsionar la recomendación. Sí se ha visto, sin embargo, el efecto retroactivo que produce en la distribución de popularidad la intervención de los recomendadores (Sharma et al 2015, Adamopoulos et al 2015): lo más popular es lo que se recomienda y, por tanto, lo que más se descubre y más probabilidades presenta de obtener un voto. Este fenómeno también se ha observado en redes sociales como Twitter, cuando los ítems que se recomiendan son los propios usuarios (Su et al 2016).

También en relación con el proceso de generación de ratings, cabe resaltar el estudio de Salganik et al (2006), que pone de manifiesto la presencia de un cierto grado de incertidumbre y azar en el orden en que se descubren los productos, incluso en ausencia de influencias externas, lo que nos lleva a cuestionar el hecho de que lo más popular sea lo que más gusta y, por tanto, una buena opción a la hora de recomendar. Otro descubrimiento llamativo de este estudio es que al incluir la influencia social, es decir, al informar a los usuarios de las opiniones de otros, no sólo se acentúa la incertidumbre sino también el sesgo de la distribución de popularidad. Otros estudios posteriores en esta línea confirman el efecto de la influencia social en el aumento de sesgos de popularidad (Wang & Wang 2014).

En el presente trabajo, sin embargo, no entramos en un análisis del proceso que da lugar a las distribuciones de popularidad. Partimos en su lugar de una distribución ya formada y buscamos identificar las propiedades de dicha distribución que determinan la efectividad de la recomendación por popularidad.

3.3.5 Popularidad resultante de fenómenos de red social

En la línea de trabajos que investigan la formación de las distribuciones de popularidad, en (Cañamares & Castells 2014) postulamos un modelo probabilístico de comportamiento de los usuarios en su comunicación con sus contactos y su interacción (rating) con ítems, viendo la generación de ratings como un proceso sujeto a fenómenos de red. En dicho trabajo analizamos además el efecto del comportamiento global resultante de la red, y las distribuciones de votos a los que este da lugar, en la recomendación y la evaluación realizadas empleando dichos votos. Si bien en aquel trabajo no entramos en un desarrollo analítico de la relación entre las distribuciones y el efecto en la recomendación, sí se identifican y caracterizan mediante simulación situaciones donde la popularidad funciona bien o mal en términos de la precisión real y observada. La comunicación en la red social se modeliza como la principal causa del descubrimiento de ítems y estudiamos distintas situaciones en cuanto a la forma de producirse dicha comunicación (si se habla más de lo relevante o de lo no relevante, por ejemplo).

Mediante una simulación del modelo, se recrean las dinámicas temporales de propagación de información en la red social que hacen que los usuarios vayan descubriendo los ítems y votándolos. Se trata por tanto de un modelo temporal de generación de ratings, lo que permite realizar una partición temporal a la hora de realizar la evaluación de los recomendadores. Cabe destacar que los recomendadores no intervienen en la propagación de información, únicamente se evalúan para comprobar su efectividad en las distintas situaciones. Más adelante en el capítulo 7 del presente trabajo se comenta en detalle las posibles derivas de la partición temporal, algunas de las cuales tienen lugar en el estudio que estamos mencionando.

El estudio (Cañamares & Castells 2014) se trata por tanto de un paso previo al presente trabajo, donde nos enfocamos en un origen concreto de la distribución de descubrimiento, que da lugar a algunas de las situaciones que se describen en este documento. En algunas de estas situaciones, por ejemplo cuando los usuarios hablan más de lo que no les gusta, se producen inversiones en la precisión de la popularidad y la recomendación aleatoria. En el presente trabajo ampliamos la perspectiva y nos abstraemos del origen de los ratings, considerando en su lugar las propiedades de la distribución resultante, la cual tomamos como punto de partida.

4. Formulación teórica

De acuerdo con lo expuesto anteriormente en la sección 1.2, el objetivo del presente trabajo es estudiar en qué medida y bajo qué circunstancias la recomendación por popularidad es una aproximación eficaz o no. En base a este objetivo, en este capítulo desarrollamos una expresión analítica general que cuantifica dicha efectividad y que permite, posteriormente, considerar y analizar distintas situaciones.

Para el desarrollo de dicha expresión, vamos a formular la efectividad como la esperanza de la métrica $P@1$, identificando para ello en primer lugar las variables de las que depende (marco probabilístico), esto es, aquellos elementos que influyen en la evaluación y cuyo valor no está fijado a priori. En base a este marco es sencillo formular posteriormente las precisiones real y observada.

4.1 Marco probabilístico

Para desarrollar una análisis formal de la efectividad de la recomendación por popularidad definimos, en primer lugar, un marco preciso en el que se concreta cómo se va a medir la eficacia de una recomendación y qué conceptos y variables se van a emplear para caracterizar distintas situaciones. A continuación detallamos y concretamos estos aspectos, que permiten definir el entorno sobre el que posteriormente vamos a realizar las distintas formulaciones y análisis teóricos.

4.1.1 Factores en la recomendación

Para estudiar la recomendación por popularidad en diversas situaciones vamos a concretar qué variables se van a emplear para caracterizar diferentes condiciones. Las variables de interés son aquellas que tengan cierta influencia en la tarea de recomendación, por lo que vamos a deducirlas analizando dicha tarea. Para ello, nos abstraemos de momento de la popularidad y contemplamos un recomendador cualquiera.

Tal y como se detalla en la sección 3.1, un recomendador es un algoritmo que a partir de unos datos de entrada acerca de interacciones entre usuarios e ítems – ratings positivos o negativos en nuestro estudio – intenta predecir los intereses de los usuarios y sugerirles nuevos productos. Las recomendaciones resultantes dependen por tanto principalmente del algoritmo recomendador y de la distribución de estos datos de entrada. Para desarrollar un análisis genérico, en el que nos abstraigamos de un conjunto de datos concreto, haremos un planteamiento de incertidumbre al respecto, en el que describiremos los datos en términos de su distribución a fin de considerar distintas situaciones. Por otra parte los propios algoritmos de recomendación contienen un cierto componente aleatorio, como es el caso en los algoritmos que utilizan valores iniciales aleatorios en un método iterativo, o cuando cualquier algoritmo tiene que desempatar entre ítems con el mismo valor de la función de ránking, o el caso más claro y obvio de la recomendación aleatoria pura.

Este factor de incertidumbre motiva el empleo de variables aleatorias. Así, si consideramos un recomendador θ, sus recomendaciones a los distintos usuarios pueden describirse como los valores posibles de una variable aleatoria, a la que vamos a denominar $\mathcal{R}$, y los datos de entrada

como una muestra concreta de una variable δ que toma valores en el conjunto de todos los datos de entrada posibles (conjunto que afortunadamente no necesitaremos definir). En esta situación, las dependencias entre dichas variables son las que se describen en forma de red bayesiana en la Figura 5.

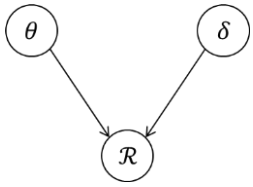

Figura 5. Red bayesiana que indica que las recomendaciones ($\mathcal{R}$) producidas por un sistema de recomendación dependen del algoritmo recomendador empleado (θ) y de los datos de entrada (δ).

Aunque en la red bayesiana el recomendador se representa como una variable aleatoria, en este estudio el valor de dicha variable viene dado y es el mismo en todas las situaciones, pues el objetivo es analizar cómo varía en dichas situaciones la efectividad de un recomendador concreto. Nos será útil tal variable no obstante para nuestra notación.

Si el recomendador está fijo, la variación entre las recomendaciones de unas situaciones y otras reside por tanto en la muestra de ratings Esta distribución depende de procesos (la generación de ratings) que pueden describirse como aleatorios, en los que intervienen tres factores fundamentales: el encuentro entre usuarios e ítems (descubrimiento), el gusto personal de los usuarios hacia los ítems (relevancia), y el comportamiento de los usuarios ante la decisión de puntuar un ítem o no. Manejaremos tres variables aleatorias correspondientes a estos tres tipos de relación entre usuarios e ítems, que comentamos a continuación. Las tres variables aleatorias son binarias y su dominio es el espacio muestral $\mathcal{U} \times \mathcal{I}$, donde $\mathcal{U}$ es el conjunto de usuarios e $\mathcal{I}$ el de ítems.

Descubrimiento

En primer lugar, para que un usuario llegue a puntuar a un cierto producto es necesario que lo conozca, que lo haya descubierto. La distribución del descubrimiento de los ítems puede llegar a ser determinante en la distribución de ratings, pues los ítems que más se descubran tendrán más posibilidades de ser puntuados y, por tanto, de alcanzar las primeras posiciones del ránking por popularidad.

Este descubrimiento puede llevarse a cabo a través de muy diversos medios (buscadores, publicidad, red social, recomendadores, etc.) y es esencialmente fortuito, pues aunque el usuario puede elegir el medio por el que informarse, no puede obviamente determinar de antemano qué ítems va a descubrir.

Pese a ello, se puede considerar una cierta tendencia a encontrar ítems que nos interesan, pues tendemos a relacionarnos con gente que tiene gustos similares a los nuestros o buscamos información en lugares que sabemos que suelen mostrar cosas que nos gustan, y nos servimos de la capacidad de herramientas como los buscadores, interfaces de navegación, o los propios recomendadores, para encontrar información que satisface nuestro interés.

El descubrimiento se representará en nuestro modelo por la variable $seen$, que vale 1 si el usuario conoce el ítem y 0 en caso contrario.

Decisión de interactuar: votación

Tras descubrir el ítem el usuario puede decidir votarlo (comprarlo, interactuar con él, etc.) o no. Esta decisión, al igual que el resultado de la votación, suele depender de si le ha gustado o no. De hecho, los datasets públicos muestran que los hábitos de los usuarios a la hora de emitir votos suelen estar sesgados hacia los casos positivos (Steck 2010).

Representaremos la acción de votar por la variable $rate$, que vale 1 si el usuario ha votado el ítem y 0 en otro caso. Es importante no confundir el valor de rating con el valor de la variable aleatoria $rate$. Esta variable indica si un usuario ha votado o no un ítem, pero no dice nada del valor de dicho voto, que puede ser de hecho positivo o negativo.

Relevancia

Tal y como se ha explicado anteriormente, tanto el descubrimiento como la decisión de votar un ítem están comúnmente condicionadas por los gustos e intereses del usuario, a los que denominamos relevancia, tomando esta denominación del campo de la Recuperación de Información (Baeza & Ribeiro 2011).

En la tarea de recomendación, la relevancia es una variable de la que el sistema únicamente conoce una parte, la manifestada a través de los votos de los usuarios. En este trabajo emplearemos el término relevancia observada para referirnos a ella, por oposición a la relevancia real que engloba también los gustos que no se han manifestado. Un recomendador emplea la relevancia observada para adivinar la relevancia real que no conoce. Esta distinción se ha de tener en cuenta a la hora de evaluar con las metodologías offline, pues estas únicamente trabajan con datos observados, tanto para recomendar como para evaluar.

Aunque se podrían considerar modelos en los que los gustos cambiaran con el tiempo en función de factores como la opinión de los amigos, las críticas, la experiencia, etc., en este estudio vamos a considerar que las opiniones de los usuarios permanecen constantes y no dependen de ningún otro factor. Establecer una relevancia variable iría más acorde con un proyecto en el que observar la formación, influencias y propagación de estados de opinión, gustos y relevancias fuera uno de los principales objetivos. En nuestro caso estamos más interesados en estudiar su efecto en el resto de factores, y por tanto introducir una relevancia cambiante sólo complicaría dicho estudio, sin aportar mayor aclaración a las preguntas que investigamos. Además de los gustos de los usuarios se podrían considerar otras influencias (amigos, críticas, etc.) pero no las consideramos por simplicidad y, de nuevo, por no ser relevantes para la cuestión específica que investigamos aquí.

Vamos a representar la relevancia (real) por la variable rel, que al evaluarse sobre un usuario u y un ítem i toma el valor 1 si el ítem resulta relevante para el usuario y 0 si no lo hace.

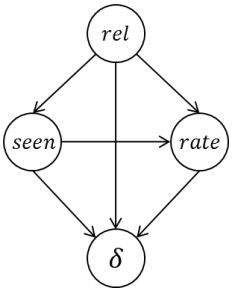

Figura 6. Red bayesiana que representa el sentido de la influencia entre las variables de descubrimiento, voto y relevancia. También se incluye la influencia que todas ellas tienen en la distribución de ratings.

Las relaciones entre las variables de descubrimiento, relevancia y voto, junto con la influencia que todas ellas tienen en la distribución de ratings, pueden observarse en la red bayesiana de la Figura 6. Tal y como hemos explicado anteriormente, consideramos que la relevancia no depende de ninguna otra variable, pero sí puede influir en el descubrimiento y la votación. Así mismo, para que un ítem se pueda votar es necesario que sea descubierto, por lo que el descubrimiento también influye en el voto. En función de si alguna de estas posibles dependencias están o no presentes tendremos unas situaciones u otras, que darán lugar a distintas distribuciones de ratings y con ellas a distintos comportamientos de los recomendadores. Así, la red final resulta de combinar las redes de las Figuras 5 y 6.

4.1.2 Factores en la evaluación

Hasta ahora, para concretar qué se entiende por popularidad y qué factores determinan su comportamiento, nos hemos abstraído de la forma en la que se va a medir su efectividad (métrica, protocolo de partición, etc.). Sin embargo, este elemento puede afectar – incluso drásticamente – tanto al resultado de la medición como, de hecho, a la propia recomendación en sí, pues determina qué datos de entrada se van a suministrar al algoritmo, así como la inclusión o exclusión de determinados ítems en el ránking de recomendación que se va a solicitar al algoritmo (Bellogín et al 2011, Said & Bellogín 2014).

Por ello, introducimos los factores relevantes de este proceso como variables de nuestro modelo, de las que depende la efectividad. En esta sección se tratará también la métrica específica con la que se va a realizar la evaluación.

4.1.2.1 Influencia de la evaluación en la recomendación

Como ya se explicó en la sección 2.1, las metodologías de evaluación que aquí consideramos – las metodologías offline – dividen los datos disponibles en un conjunto de entrenamiento, que se da como entrada a los algoritmos de recomendación, y un conjunto de test, que se oculta a los algoritmos y se emplea para evaluar la capacidad de acierto de los mismos.

Bajo esta metodología, la división en entrenamiento y test puede ser clave a la hora de determinar qué ítems se recomiendan. Por ello, en la red bayesiana de la Figura 5 consideramos también la influencia en las recomendaciones del protocolo de partición, al que nos referiremos como π. La nueva situación se muestra en la Figura 7.

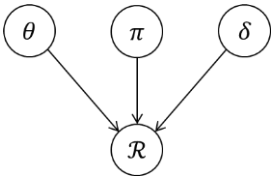

Figura 7. Ampliación de la red bayesiana de la Figura 5 considerando también la influencia del protocolo de evaluación en el propio resultado de la recomendación.

Asociada al protocolo de partición es útil considerar también otra variable aleatoria $\mathcal{S}$ que represente la partición concreta generada a partir del protocolo π. Estrictamente hablando, es de esta última variable aleatoria de la que depende realmente la recomendación, pero se asume únicamente la dependencia del protocolo para facilitar la formulación analítica posterior. La variable $\mathcal{S}$, sin embargo, es necesaria para cuantificar el resultado de la evaluación, como se verá más adelante. Así mismo, también apoya la definición de dos nuevas variables aleatorias binarias: las variables $training$ y $test$, que definimos en el espacio muestral $\mathcal{U} \times \mathcal{I} \times \mathcal{S}$, donde $\mathcal{S}$ es el conjunto de todas las posibles particiones. La variable $training$ toma el valor 1 si al evaluarse sobre un usuario u, un ítem i y una partición s concretos el usuario ha votado el ítem y el voto se encuentra en el conjunto de entrenamiento especificado por la partición s. Análogamente, la variable $test$ vale 1 si el voto se encuentra en el conjunto de test.

4.1.2.2 Métrica de evaluación

Nuestro análisis formal de la efectividad (en términos de acierto) de la recomendación se basa en una formulación de la esperanza de una métrica que valore dicha efectividad. Así pues nos referiremos al valor de la métrica en cuestión como una variable aleatoria M, que al igual que la variable θ cumplirá un fin de notación y expresión de relaciones fundamentalmente. Respecto a las relaciones con otras variables, el valor de la métrica depende en primer lugar, de las recomendaciones $\mathcal{R}$ generadas por el recomendador y de los datos de entrada δ. Esta dependencia es obvia, pues la métrica es función directa de $\mathcal{R}$ (los ránkings) y δ (de donde se extraen los juicios de relevancia). También la división en entrenamiento y test ($\mathcal{S}$) es un factor influyente, pues determina con qué valores se va a realizar la evaluación (conjunto de test).

En la Figura 8 mostramos la red bayesiana extendida con las variables que intervienen en la evaluación.

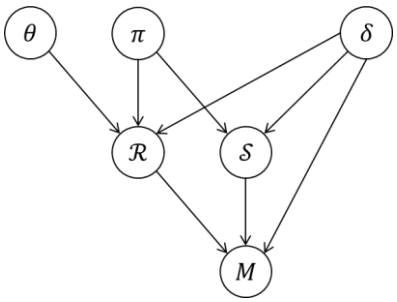

Figura 8. Red bayesiana que representa las conexiones entre las distintas variables consideradas en el marco de estudio.

Respecto a la métrica específica de evaluación, tal y como explicábamos en la sección 3.1.2 existen numerosas métricas que permiten evaluar lo efectivo que es un recomendador (Recall, Precisión, nDCG, etc.). En este trabajo empleamos la métrica de precisión por ser una de las más utilizadas en el área y porque por su sencillez es más tratable a la hora de formularla en términos de esperanzas y probabilidades.

La precisión $P@k$ de una recomendación ofrecida a un usuario concreto se define como la tasa de aciertos en las primeras k posiciones de la recomendación. Cuando se consideran varios usuarios, cada uno con su correspondiente recomendación, la precisión global es el promedio de la precisión de las recomendaciones individuales. Así, si R_u^k denota las primeras k posiciones de la recomendación ofrecida al usuario u, se tiene que:

$$P@k = \underset{u \in \mathcal{U}}{\text{avg}} \, |\{i \in R_u^k | \text{ a } u \text{ le gusta } i\}|/k$$

Para simplificar, consideramos $P@1$, pues por su sencillez admite mejor un tratamiento analítico, ya de por sí bastante complejo, sin dejar por ello de ser representativa como métrica de evaluación. Contrastaremos no obstante en el capítulo 8 el comportamiento con $P@10$, y comprobaremos, como se verá, que coincide.

Una vez establecido el marco de estudio y las variables de interés, es posible desarrollar la expresión analítica que permite cuantificar el valor de la precisión en función de dichas variables. En la siguiente sección exponemos el desarrollo formal de dicha expresión para un caso general, dejando para posteriores capítulos la consideración de casos particulares.

4.2 Formulación general

Si bien nuestro objetivo concreto es cuantificar la precisión de la popularidad en ciertas circunstancias y compararla con la del recomendador aleatorio, vamos a partir de una formulación más general aplicable a escenarios más genéricos (otras métricas, otros recomendadores, etc.), que no particularizaremos al de la precisión de la popularidad hasta que no sea necesario. Vamos pues a desarrollar una expresión que cuantifique la eficacia de un recomendador al ser evaluado según las metodologías offline. Respecto a la métrica de evaluación, por ahora podemos abstraernos de la precisión y seguir considerando una cierta función M genérica.

Dados un recomendador, unos datos y un protocolo de partición, las recomendaciones $\mathcal{R}$ y la partición $\mathcal{S}$ pueden variar de unas ejecuciones a otras, y con ellos el resultado de la métrica M. Por ello, no hablamos de un valor exacto de la métrica sino que consideramos la esperanza de dicha métrica promediando sobre todas las posibles particiones y recomendaciones[5].

$$\mathbb{E}[M|\theta, \pi, \delta] = \sum_{\substack{s \in \mathbb{S} \\ \mathbf{R} \in \sigma(\mathcal{I})^m}} M(\mathbf{R}, s, \delta)\, p(\mathbf{R}|\theta, \pi, \delta)\, p(s|\pi, \delta) \qquad (1)$$

En la expresión anterior, $\mathbb{S}$ es el conjunto de todas las posibles particiones y $\sigma(\mathcal{I})^m$ representa el conjunto de todas las posibles recomendaciones (las permutaciones $\sigma(\mathcal{I})$ del conjunto de ítems "elevadas al número de usuarios", es decir $\mathbf{R}$ incluye una permutación de ítems – una recomendación, un ránking – $R \in \sigma(\mathcal{I})$ por usuario), donde cada recomendación estaría compuesta a su vez por los ránkings ofrecidos a los distintos usuarios. Cabe destacar que cada uno de estos ránkings es una ordenación de los todos los ítems, por lo que más adelante habrá

[5] Abuso de notación: $p(\mathbf{R}|\theta, \pi, \delta)\, p(s|\pi, \delta) \equiv p(\mathcal{R} = \mathbf{R}|\theta, \pi, \delta)\, p(\mathcal{S} = s|\pi, \delta)$. Este tipo de notación es común en la literatura de Recuperación de Información, y se va a mantener a lo largo del documento por simplificación. Así mismo, en este punto se emplea la red bayesiana para descomponer las probabilidades de la partición y las recomendaciones de la siguiente forma: $p(\mathbf{R}, s|\theta, \pi, \delta) = p(\mathbf{R}|\theta, \pi, \delta)\, p(s|\pi, \delta)$

que considerar la posibilidad de que algunos de esos ítems tengan que ser excluidos debido a que ya se encuentren en entrenamiento. En dicho caso el usuario ya los conoce y no deben volver a recomendarse.

Respecto a la expresión $M(\mathbf{R}, s, \delta)$, se refiere al valor de la métrica resultante de evaluar las recomendaciones **R** cuando los datos δ han sido divididos según indica la partición s. Para seguir avanzando asumimos que M es una métrica que se evalúa usuario a usuario y posteriormente se promedia para obtener el valor global de todas las recomendaciones, como es el caso de la precisión:

$$M(\mathbf{R}, s, \delta) = \frac{1}{m}\sum_{u} M_u(u, R_u, s, \delta)$$

En la expresión anterior, M_u es el valor de la métrica para el usuario u. Este valor ya no depende de todas las recomendaciones sino únicamente de la correspondiente a dicho usuario, a la que denominamos R_u. Es por ello que la red bayesiana que refleja las dependencias incluyendo esta variable varía ligeramente de la global, tal y como mostramos en la Figura 9. En ella, el usuario también se considera como una variable aleatoria de la que claramente depende el valor de M_u.

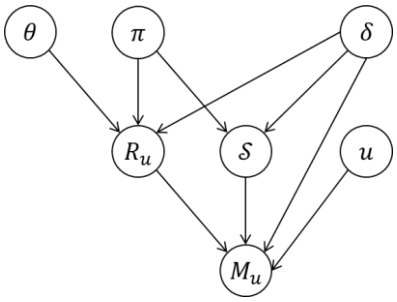

Figura 9. Red bayesiana que representa las conexiones entre las distintas variables que influyen en el valor de la métrica para un usuario concreto.

Sustituyendo en la fórmula 1 inicial el valor de $M(r, s, \delta)$ y llevando el sumatorio en la partición todo lo internamente posible se tiene que:

$$\begin{aligned}\mathbb{E}[M|\theta, \pi, \delta] &= \sum_{\mathbf{R}\in\sigma(\mathcal{I})^m} p(\mathbf{R}|\theta, \pi, \delta)\ \frac{1}{m}\sum_{u}\sum_{s\in\mathbb{S}} M_u(u, R_u, s, \delta)\ p(s|\pi, \delta) \\ &= \sum_{\mathbf{R}\in\sigma(\mathcal{I})^m} p(\mathbf{R}|\theta, \pi, \delta)\ \frac{1}{m}\sum_{u} \mathbb{E}[M_u|u, R_u, \pi, \delta] \qquad (2)\end{aligned}$$

De esta forma, conseguimos abstraernos de una partición concreta y consideramos la esperanza de M_u sobre todas las posibles particiones: $\mathbb{E}[M_u|u, R_u, \pi, \delta]$. Cabe destacar que una vez conocida la recomendación concreta R_u, esta esperanza ya no depende del recomendador. El motivo de considerar la esperanza en lugar del valor para cada partición concreta, es que más adelante va a ser posible estimar directamente el valor de esta esperanza a partir del protocolo de partición, cuando dicho protocolo cumple una serie de características.

El planteamiento desarrollado hasta este punto es válido para cualquier métrica que se evalúe usuario a usuario. Para continuar con el desarrollo, sin embargo, vamos a centrarnos en la precisión del primer ítem recomendado: $P@1$.

Así, la precisión $P@1(u, R_u, s, \delta)$ es una función que vale 1 si el primer ítem de R_u es un acierto para u y 0 en caso contrario. Más adelante analizaremos los dos tipos de precisión (real

y observada) que surgen al considerar qué se entiende por acierto. Para el siguiente paso, sin embargo, es suficiente con saber que la precisión – en todas sus variantes – es una función binaria y, por tanto, su esperanza es precisamente la probabilidad de que valga 1. Así pues tenemos[6]:

$$\mathbb{E}[P@1|u, R_u, \pi, \delta] = p(P@1(u, R_u, \pi, \delta) = 1) = p(P@1|u, R_u, \pi, \delta)$$

Como ya adelantamos anteriormente, R_u es una ordenación concreta de todos los ítems del sistema, incluidos aquellos que u ya ha votado y cuyos votos han ido a parar al conjunto de entrenamiento. Dichos ítems ya son conocidos por u y no tiene sentido volver a recomendarlos, por lo que deben ser excluidos. Consecuentemente, la precisión se evalúa sobre el primer ítem recomendable, esto es, sobre el primer ítem de R_u que no ha sido votado por el usuario en el conjunto de entrenamiento. Este ítem es una función τ del ránking R_u, del usuario u, del protocolo de partición π y de los datos δ, sin embargo por simplicidad denotaremos a dicho elemento como $\mathbb{I}_u \coloneqq \tau(u, R_u, \pi, \delta)$.

La probabilidad de que la precisión valga 1 es por tanto la probabilidad de que este primer ítem $\mathbb{I}$ sea un acierto.

$$\mathbb{E}[P@1|u, R_u, \pi, \delta] = p(\mathbb{I}_u \; sea \; un \; acierto \; para \; u|u, R_u, \pi, \delta)$$

Para simplificar y evitar arrastrar términos que compliquen la notación, de ahora en adelante vamos a eliminar las variables π y δ de todas las condiciones. Sin embargo, es importante recordar que realmente estas variables están fijadas pues más adelante habrá que volver sobre ellas, en particular para considerar distintos protocolos de partición.

En este punto abordamos el análisis de qué se considera un acierto y desarrollamos las precisiones real (P) y observada ($\bar{P}$) por separado.

4.2.1 Precisión observada

Para la precisión observada, a la que denotaremos como $\bar{P}$, el ítem $\mathbb{I}$ es un acierto si es relevante para u y si dicha relevancia ha sido observada en el conjunto de test, esto es, si se tiene un voto de u sobre dicho ítem en test. La probabilidad de que $\mathbb{I}_u$ sea un acierto es, por tanto, la probabilidad de que sea relevante y este en test.

$$\mathbb{E}[\bar{P}@1|u, R_u] = p(test(\mathbb{I}_u), rel(\mathbb{I}_u)|u, R_u)$$

[6] Doble abuso de notación que mantenemos a lo largo del documento:

- Si X es una variable aleatoria binaria (como rel o $test$) se escribirá $p(X)$ en lugar de $p(X = 1)$.
- Cuando en una probabilidad las variables aleatorias que aparezcan tengan los mismos argumentos estos se colocaran en la parte condicionada.
- Ejemplos:
  - $p\big(test(u, i, s), rel(u, i)\big) \equiv p(test, rel|u, i, s)$
  - $p\big(test(u, i_1, s), rel(u, i_2)\big) \equiv p(test(i_1), rel(i_2)|u, s)$
  - $p\big(P@1(u, R_u, \pi, \delta)\big) \equiv p(P@1|u, R_u, \pi, \delta)$

  Si uno de los argumentos de la variable no aparece se asumirá que puede tomar cualquier valor en ese campo. Por ejemplo, la probabilidad $p(test|u, i)$ donde falta el valor de la variable partición se interpreta como la probabilidad de que un posible voto de u a i se encuentre en test, sea cual sea la partición.

Ahora bien, R_u es una ordenación concreta del conjunto de ítems, pongamos $R_u = \{i_1, \dots, i_n\}$, por lo que podemos descomponer la probabilidad anterior particionando por los eventos correspondientes a que cada uno de los ítems de R_u sea el primer elemento recomendable.

$$\mathbb{E}[\bar{P}@1|u, R_u] = \sum_{k=1}^{n} p(test(i_k), rel(i_k), \mathbb{I}_u = i_k|u, R_u)$$

Para que el ítem i_k sea el primero recomendable es necesario que todos los anteriores estén en el conjunto de entrenamiento pero que él no lo esté, por lo que el evento $\mathbb{I}_u = i_k$ podría reescribirse de la siguiente forma.

$$(\mathbb{I}_u = i_k) = \left(\neg training(i_k), training(i_1), \dots, training(i_{k-1})\right)$$

Y al sustituir en la esperanza de la precisión observada por usuario tenemos:

$$\mathbb{E}[\bar{P}@1|u, R_u] = \sum_{k=1}^{n} p(test(i_k), rel(i_k), training(i_1), \dots, training(i_{k-1})|u, R_u) \qquad (3)$$

Donde hemos eliminado la condición $\neg training(i_k)$ porque está implícita en el evento $test(i_k)$.

4.2.2 Precisión real

Mientras que la precisión observada representa el acierto medido del sistema, la precisión real, denotada por *P*, calcula la tasa de recomendaciones acertadas teniendo en cuenta la información completa de relevancia. Su cómputo presupondría conocimiento completo de los usuarios, independientemente de que estos hayan sido manifestados o no a través de sus votos, lo cual es por lo general inviable, pero ello no nos impide razonar sobre este concepto, o aproximarnos a él. Así, en este caso la probabilidad de que $\mathbb{I}_u$ sea un acierto es simplemente la probabilidad de que sea relevante.

$$\mathbb{E}[P@1|u, R_u, \pi, \delta] = p(rel(\mathbb{I}_u)|u, R_u)$$

Repitiendo los pasos desarrollados en el caso de la precisión observada – particionar por el suceso $\mathbb{I}_u = i_k$ y sustituir adecuadamente dicho evento – la esperanza de la precisión real por usuario presenta la siguiente forma:

$$\mathbb{E}[P@1|u, R_u] = \sum_{k=1}^{n} p(rel(i_k), \mathbb{I}_u = i_k|u, R_u)$$

$$= \sum_{k=1}^{n} p(\neg training(i_k), rel(i_k), training(i_1), \dots, training(i_{k-1})|u, R_u) \qquad (4)$$

Vemos que la formulación de las precisiones observada y real se asemejan en estructura, y difieren en el término $test(i_k)$ para la observada, frente a $\neg training(i_k)$ para la real. Puesto que el primer evento contiene al segundo, podemos confirmar que se cumple siempre $\bar{P} \leq P$, lo cual ya era conocido (la precisión observada es una subestimación de la real, ver Herlocker 2004) y corresponde con el sentido común.

Una vez introducida esta formulación general de la efectividad de un recomendador vamos a pasar a estudiar su comportamiento en función de distintos factores. En los dos siguientes capítulos analizamos la influencia de la distribución de popularidad y del descubrimiento,

respectivamente, en situaciones más particulares que atañen a los recomendadores aleatorio y popularidad.

5. Influencia de la distribución de popularidad

Como ya introdujimos en la sección 4.1, asumiendo un recomendador y un protocolo de partición concretos, las distintas situaciones que pueden producirse al recomendar vienen caracterizadas por la distribución de los datos de entrada δ. Estos datos de entrada pueden a su vez representarse como el producto de la interacción entre las variables rel, $rate$ y $seen$.

Estudiar la eficacia de la recomendación en distintas situaciones permite por tanto dos aproximaciones: considerar que las situaciones vienen descritas por una distribución concreta de los datos de entrada δ o por una relación entre las variables rel, $rate$ y $seen$ que genera dichos datos. En este capítulo desarrollamos la primera aproximación, dejando para el siguiente capítulo la caracterización mediante las dependencias entre las tres variables.

Considerar fijos los datos de entrada δ es un planteamiento que representa con mayor naturalidad el conocimiento que se tiene en una situación típica de evaluación offline – donde no se suelen conocer las distribuciones completas de las variables rel y $seen$ – y resulta en un análisis que se presta a ser contrastado con mayor facilidad, pues únicamente es necesario evaluar el resultado de las fórmulas sobre distintos conjuntos de datos. El desconocimiento del valor real de la relevancia complica además calcular la precisión real en estas situaciones, por lo que en este primer planteamiento nos reduciremos al caso de la precisión observada.

Cabe destacar que para este desarrollo consideramos que no se conoce el valor concreto de δ – quién ha votado qué – sino únicamente sus distribuciones de popularidad – cuantos votos (relevantes y no relevantes) ha realizado cada usuario y cuantos ha recibido cada ítem. El motivo de caracterizar los datos de entrada por sus distribuciones de popularidad es que resulta lógico pensar que pueden ser la característica de dichos datos que más influye en el comportamiento del recomendador por popularidad. Además, esta asunción permite diferenciar el fenómeno de popularidad del resto de posibles dependencias que puedan existir ente los datos, para así poder estudiar su efecto de forma aislada.

En este capítulo, por tanto, el objetivo es doble: se trata, por un lado, de contrastar empíricamente que las fórmulas predicen correctamente el valor de la precisión observada en una situación típica de evaluación offline y, por otro, de estudiar el efecto que la distribución de popularidad tiene en dicha precisión. En base a estos objetivos, consideramos un escenario con los dos recomendadores que nos interesan – aleatorio y popularidad – que se ejecutan sobre unos datos de entrada de los que se conoce su distribución de popularidad y que son divididos mediante una partición aleatoria.

5.1 Formulación analítica

Para desarrollar la ecuación 3 de la precisión observada en la situación concreta que nos interesa tenemos en cuenta que al establecer únicamente la distribución de popularidad, la distribución de ratings δ no es fija y, por ello, para calcular los valores de las variables de relevancia y votación que aparecen en las fórmulas es necesario estimarlos a partir de dicha

distribución de popularidad. Respecto a la variable descubrimiento, una vez fijada la distribución de popularidad de δ, no tiene influencia en el resultado, por lo que en este capítulo la vamos a ignorar y considerar la situación representada por la red bayesiana de la Figura 10.

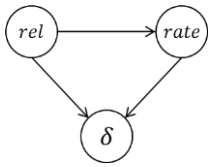

Figura 10. Red bayesiana que relaciona las variables de votación y relevancia con la distribución de ratings.

Respecto al protocolo de partición, en una partición aleatoria típica cada voto tiene la misma probabilidad de ser asignado al conjunto de entrenamiento, probabilidad que denotamos como ρ. Este valor se puede formular como la probabilidad de estar en training una vez que se sabe que existe un voto:

$$\rho = p(training|rate, i, u)$$

Dicha probabilidad coincide con el porcentaje de votos de entrenamiento frente a los de test, motivo por el cual se le suele denominar *tasa de entrenamiento*.

El hecho de que sea el mismo valor para todos los votos implica que la asignación a entrenamiento (y a test) sigue una distribución binomial. Más adelante consideraremos particiones aleatorias multinomiales en los que en función del ítem al que haga referencia el voto se tiene una probabilidad distinta ρ_i de estar en entrenamiento.

Para reducir las fórmulas en el caso de una partición aleatoria binomial tenemos en cuenta que se cumplen las siguientes igualdades[7]:

$$p(training(i), \dots |u, R_u) = \rho\, p(rate(i), \dots |u, R_u)$$
$$p(test(i), \dots |u, R_u) = (1-\rho)\, p(rate(i), \dots |u, R_u)$$

Donde los puntos suspensivos pueden ser cualquier lista de variables aleatorias.

Aplicando sucesivamente estas igualdades, la fórmula 3 para la precisión observada resulta de la siguiente forma:

$$\mathbb{E}[\bar{P}@1|u, R_u] = \frac{(1-\rho)}{\rho} \sum_{k=1}^{n} \rho^k p(rate(i_k), rel(i_k), rate(i_1), \dots, rate(i_{k-1})|u, R_u) \quad (5)$$

Hasta el momento hemos considerado un recomendador genérico θ, lo que impide desarrollar en más detalle la fórmula, pues no podemos hacer ninguna afirmación sobre los ítems que forman R_u. A continuación estudiamos los casos concretos del recomendador aleatorio y de popularidad.

[7] Las variables $training(u, i)$ y $test(u, i)$ sólo pueden valer 1 si i ha sido votado por u, es decir, si $rate(u, i)$ también vale 1. Esto implica que $p\big(training(u, i)\big|\neg rate(u, i)\big) = 0$ y por tanto:

$$p(training(i), \dots |u, R_u) = p(training(i), rate(i), \dots |u, R_u)$$

Aplicando la regla de la probabilidad condicionada se tiene que

$$p(training(i), rate(i), \dots |u, R_u) = p(training(i), \dots |rate(i), u, R_u)\, p(rate(i), \dots |u, R_u)$$

En una partición aleatoria binomial $p(training(i), \dots |rate(i), u, R_u) = \rho$ por lo que se llega a

$$p(training(i), \dots |u, R_u) = \rho\, p(rate(i), \dots |u, R_u)$$

Análogamente para la variable $test$.

5.1.1 Recomendador aleatorio

El recomendador aleatorio elige con probabilidad uniforme los ítems que va a recomendar. Esto implica que todos los ránkings son igual de probables y permite desarrollar la fórmula sin necesidad de fijar uno en concreto. Al quitar el ránking R_u, dejamos el recomendador como parámetro de la esperanza por usuario.

$$\mathbb{E}[\bar{P}@1|u, R_u] \sim \mathbb{E}[\bar{P}@1|u, \theta]$$

Lo que esta notación indica es que se puede calcular la esperanza por usuario conociendo únicamente el recomendador, sin necesidad de considerar cada posible ránking por separado. Así, en el desarrollo de $\mathbb{E}[\bar{P}@1|u, \theta]$ los ítems i_k en lugar ser los ítems del ránking R_u son los ítems generados por el recomendador θ. Al ser el recomendador aleatorio esto quiere decir que pueden ser cualquier ítem dentro de los que no están en training o ya han sido recomendados.

Esta simplificación implica que en la fórmula 2 de la precisión global ya no es necesario considerar todos los ránkings posibles – esta consideración está implícita en la esperanza de la métrica por usuario – y es suficiente con promediar sobre todos los usuarios.

$$\mathbb{E}[\bar{P}@1|\theta] = \frac{1}{m}\sum_{u} \mathbb{E}[\bar{P}@1|u, \theta] \tag{6}$$

Para cuantificar la esperanza de la precisión observada (ecuación 5) debemos calcular el valor de la expresión $p(rate(i_k), rel(i_k), rate(i_1), \dots, rate(i_{k-1})|u, \theta)$. Para ello, la descomponemos según la regla de la probabilidad condicionada:

$$\begin{aligned} p(rate(i_k), rel(i_k), rate(i_1), \dots, rate(i_{k-1})|u, \theta) \\ = p(rate(i_k), rel(i_k)|rate(i_1), \dots, rate(i_{k-1}), u, \theta) \\ \cdot \prod_{j=1}^{k-1} p\big(rate\big(i_j\big)\big|rate(i_1), \dots, rate\big(i_{j-1}\big), u, \theta\big) \end{aligned}$$

Para calcular $p\big(rate\big(i_j\big)\big|rate(i_1), \dots, rate\big(i_{j-1}\big), u, \theta\big)$ tenemos en cuenta que el recomendador elije los i_j de forma uniforme – dentro de los que no están en training o ya han sido recomendados – y, por tanto, se puede emplear la clásica fórmula de casos favorables entre casos posibles.

Dado que estamos considerando $\bar{P}@1$ – la precisión en la primera posición – no hay ítems ya recomendados, por lo que para calcular el número de casos posibles únicamente se excluyen los que ya se sabe que están en entrenamiento. Respecto al número de casos favorables, estos son el número de votos del usuario menos los $j-1$ votos que ya se saben corresponden a unos ítems concretos.

$$p\big(rate\big(i_j\big)\big|rate(i_1), \dots, rate\big(i_{j-1}\big), u, \theta\big) = \frac{|u| - j + 1}{n - j + 1}$$

En la expresión anterior $|u|$ denota el número de votos de u y n el número total de ítems. Cabe destacar en este punto que lo que estamos haciendo es estimar el valor de la probabilidad anterior a partir de la distribución de popularidad de δ, pues el valor exacto se desconoce ya que δ puede ser cualquier distribución de ratings que presente la distribución de popularidad fijada.

En el caso de la expresión $p(rate(i_k), rel(i_k)|rate(i_1), \dots, rate(i_{k-1}), u)$ empleamos nuevamente la regla de la probabilidad condicionada para calcular su valor[8]:

$$(rate(i_k), rel(i_k)|rate(i_1), \dots, rate(i_{k-1}), u, \theta)$$
$$= p(rel(i_k)|rate(i_k), rate(i_1), \dots, rate(i_{k-1}), u, \theta)$$
$$\cdot\ p(rate(i_k)|rate(i_1), \dots, rate(i_{k-1}), u, \theta) = \frac{|u^{rel}|}{|u|} \cdot \frac{|u| - k + 1}{n - k + 1}$$

Sustituyendo todos estos valores en la ecuación 5 y esta a su vez en la fórmula 6 obtenemos que el valor esperado de la precisión observada del recomendador aleatorio es:

$$[\bar{P}@1|\theta = rnd] = \frac{(1-\rho)}{m \cdot \rho} \sum_u \sum_{k=1}^{n} \frac{|u^{rel}|}{|u|} \prod_{j=1}^{k} \rho \frac{|u| - j + 1}{n - j + 1} \qquad (7)$$

5.1.2 Popularidad total y relevante

Aunque el desarrollo que exponemos en esta sección lo vamos a contrastar empíricamente con los recomendadores de popularidad total y popularidad relevante, cabe destacar que el análisis es válido para cualquier otro recomendador que cumpla las hipótesis que se van a explicar a continuación.

Para desarrollar la fórmula de la precisión observada que se detalla en la ecuación 5 es necesario calcular el valor de la expresión

$$p(rate(i_1), \dots, rate(i_{k-1})|u, R_u)$$

Fijado un ránking de ítems R_u y un usuario u concretos, no es posible calcular de forma genérica el valor de esta probabilidad conociendo únicamente cuantos votos ha recibido cada ítem y cuantos votos ha realizado el usuario.

Este problema es equivalente a conocer la probabilidad de que dos nodos de un grafo estén conectados dado el grado de cada uno. Así, podemos interpretar a los usuarios e ítems como nodos y a los votos como aristas que los conectan, con la particularidad de que únicamente puede existir una arista entre un nodo que representa un usuario y otro que representa un ítem. Este tipo de grafos en los que los nodos están divididos en dos grupos y las aristas únicamente pueden conectar nodos de distinto grupo se denominan grafos bipartitos. En esta situación, establecer el número de votos de cada ítem / usuario (distribución de popularidad) es equivalente a fijar el grado de cada nodo. De igual forma, calcular la probabilidad de que exista un voto ente un usuario e ítem concretos - $p(rate|u, i)$ – es preguntar sobre la probabilidad de que haya una arista entre los nodos que los representan, conociendo los grados de cada nodo y la densidad total del grafo. El problema se puede formular también en términos de la probabilidad de que, en una matriz binaria, haya en un 1 en una celda de la que se conoce la suma de la fila y la columna en la que se encuentra.

[8] En la última igualdad se asume que el hecho de que un ítem tenga un voto no afecta a la posibilidad de relevancia de otro ítem distinto.

$$p(rel(i_k)|rate(i_k), rate(i_1), \dots, rate(i_{k-1}), u, \theta) = p(rel(i_k)|rate(i_k), u, \theta) = \frac{|u^{rel}|}{|u|}$$

Existen en la literatura aproximaciones y acotaciones de esta probabilidad (Greenhill et al 2006, Canfield et al 2008, Barvinok 2010) bajo ciertas hipótesis que en nuestro planteamiento no tienen por qué cumplirse. De igual forma también se han desarrollado algoritmos para aproximar dicho valor (Golshan et al 2013) lo cual se aleja de nuestro objetivo que es presentar una fórmula analítica cerrada.

En el caso del recomendador aleatorio, podíamos simplificar la expresión al considerar que el ránking R_u no estaba fijado y podía ser uno cualquiera. Para recomendadores más complejos en los que no todos los ránkings tienen la misma probabilidad no es posible realizar esta asunción. Por ello, consideramos otra hipótesis más realista que simplifique la fórmula: la independencia del usuario.

La independencia del usuario conlleva la asunción de que la precisión de un ránking es igual para todos los usuarios. Dado que, bajo las hipótesis que estamos considerando, los usuarios únicamente se caracterizan por el número de votos que han realizado, lo que estamos asumiendo realmente es que lo activo que sea un usuario no influye en la precisión de un ránking ofrecido a dicho usuario.

En los casos de los recomendadores no personalizados esta hipótesis no es descabellada, ya que no emplean los votos particulares del usuario a la hora de hacerle recomendaciones y, en la mayoría de casos, R_u es el mismo para todos los usuarios. Es cierto que puede haber una posible influencia del número de votos en las probabilidades de tener un voto en test y con ello en el valor de la precisión, pero consideramos esta influencia poco significativa en comparación con el resto de variables.

Por tanto, el desarrollo que sigue es válido para recomendadores no personalizados que ofrecen el mismo ránking R a todos los usuarios. Es importante recordar en este punto que el ránking R depende del protocolo de partición porque se realiza empleando únicamente los votos de entrenamiento y, a priori, varios ránkings podrían ser posibles – uno para cada posible partición. Sin embargo, por simplicidad asumimos que el protocolo mantiene en los conjuntos de entrenamiento y test la misma distribución original de los datos. Esto implica que el ránking resultante en cualquier partición es el mismo que si se realizara empleando todos los datos, lo cual hace que sea único.

La existencia de un único ránking permite simplificar la ecuación original 2 para la precisión global del sistema, pues el resto de ránkings cumplen que su probabilidad $p(R|\theta, \pi, \delta)$ es 0:

$$\mathbb{E}[P@1|\theta] = \frac{1}{m}\sum_{u} \mathbb{E}[P@1|u, R]$$

En este punto empleamos la hipótesis de independencia del usuario, que se traduce en la siguiente estimación para la precisión por usuario:

$$\mathbb{E}[P@1|u, R] \approx \mathbb{E}[P@1|R]$$

Y, por tanto, la suma por usuario se puede simplificar llegando a que la precisión global del recomendador es simplemente la precisión del ránking que genera.

$$\mathbb{E}[P@1|\theta] \approx \mathbb{E}[P@1|R]$$

En el caso específico que nos preocupa, el de las popularidades total y relevante, basta con considerar que el ránking que se está estudiando es el generado por dichos recomendadores, es

decir, que los ítems están ordenados según el número de votos y votos relevantes que tiene cada uno. La asunción de que el protocolo mantiene la distribución original en el conjunto de entrenamiento se traduce, en este caso, en que el orden de los ítems según sus votos en el conjunto de entrenamiento es el mismo que si se ordenan considerando todos sus votos, tanto los de entrenamiento como los de test. Esta asunción es bastante razonable para el caso de una partición aleatoria ya que, en principio, cuantos más votos tenga un ítem, mayor será la cantidad que pueda ir a parar a entrenamiento. Sin embargo, cuando la distribución de popularidad de los ítems es muy uniforme, las inversiones en el orden derivadas de que la varianza en la partición es mayor que la varianza de los datos pueden llegar a ser bastante probables, como veremos más adelante en la sección 7.3.

Calcular la precisión observada de un recomendador que cumple las hipótesis anteriores se reduce, por tanto, a calcular la precisión del ránking que genera. Así, dado un ránking (R) se trata de calcular el valor de la siguiente expresión:

$$\mathbb{E}[\bar{P}@1|R] = \frac{(1-\rho)}{\rho}\sum_{k=1}^{n}\rho^k p(rate(i_k), rel(i_k), rate(i_1), \dots, rate(i_{k-1})|R)$$

Al asumir la independencia del usuario, el problema se reduce a calcular la probabilidad de que una serie de ítems sean votados por un mismo usuario, sea cual sea este. Esta probabilidad se puede calcular a partir del número de usuarios que han votado todos esos ítems en el conjunto total de datos. La condición de relevancia del ítem i_k se resuelve considerando para dicho ítem únicamente los usuarios que lo han votado como relevante, conjunto al que denotamos como i_k^{rel}. Así, empleando un cierto abuso de notación en el que i_j representa tanto al ítem que ocupa la posición j del ránking como al conjunto de usuarios que han votado dicho ítem se tiene que:

$$p(rate(i_k), rel(i_k), rate(i_1), \dots, rate(i_{k-1})|R) = \frac{\left|i_1 \cap \dots \cap i_k^{rel}\right|}{m}$$

Por lo que el resultado de la precisión observada del recomendador θ es

$$\mathbb{E}[\bar{P}@1|R] = \frac{(1-\rho)}{m\cdot\rho}\sum_{k=1}^{n}\rho^k\left|i_1 \cap \dots \cap i_k^{rel}\right| \quad (8)$$

En el caso específico de las popularidades total y relevante, para calcular este valor basta con ordenar todos los ítems en función del número de votos y votos relevantes que tienen – en orden descendiente – y sumar tal y como indica el sumatorio. Cabe destacar que para calcular el valor de esta fórmula es necesario asumir que también se conoce el valor de $\left|i_1 \cap \dots \cap i_k^{rel}\right|$, esto es, el tamaño de las intersecciones entre los conjuntos de usuarios que han votado cada ítem.

5.2 Confirmación empírica

Para comprobar la corrección de las fórmulas anteriores hemos ejecutado los tres recomendadores – aleatorio, popularidad total y popularidad relevante – sobre varios conjuntos de datos públicos frecuentemente utilizados en el área de los sistemas de recomendación. Concretamente hemos empleado los conjuntos de MovieLens, Last.fm y Netflix descritos en la sección 2.

En la Tabla 2 se muestran las precisiones $\bar{P}@1$ empíricas y teóricas resultantes de evaluar cada recomendador sobre cada uno de los conjuntos anteriores. La precisión empírica la hemos obtenido dividiendo los datos mediante una partición aleatoria de parámetro $\rho = 0.8$, una de las tasas de entrenamiento más empleadas en la literatura. La precisión teórica se corresponde con el valor de la ecuación 7, para el recomendador aleatorio, y de la ecuación 8, para las popularidades total y relevante.

Se observa que los resultados teóricos aproximan a los empíricos en más de dos cifras significativas, lo cual confirma que las hipótesis asumidas en la formulación analítica – independencia del usuario y mantenimiento de la distribución original en los datos de entrenamiento y test – son correctas o producen desviaciones inapreciables.

Cabe destacar que en el caso de Last.fm todos los votos son positivos – representan escuchas de canciones por parte de los usuarios – por lo que popularidad total y relevante coinciden.

		MovieLens		**Last.fm**		**Netflix**	
		Empírica	**Teórica**	**Empírica**	**Teórica**	**Empírica**	**Teórica**
Recomendador	Aleatorio	0.005609	0.005606	0.001038	0.001040	0.001391	0.001372
	Pop. total	0.209144	0.208823	0.410267	0.415935	0.128555	0.128761
	Pop. relevante	0.209921	0.210022	0.410267	0.415935	0.146581	0.146658

Tabla 2. Precisión $\bar{P}@1$ empírica y teórica de los recomendadores aleatorio y popularidad al ejecutarlos sobre los conjuntos de datos públicos: MovieLens, Last.fm y Netflix, empleando un protocolo de partición aleatorio de parámetro $\rho = 0.8$.

5.3 Comparativa: aleatorio vs. popularidad

A la hora de realizar afirmaciones sobre la efectividad de un algoritmo – decir si es alta o baja, por ejemplo – lo propio es compararla con la de algún sistema de referencia del que ya tengamos alguna noción de lo bueno o malo que es. En este caso, el sistema de referencia que consideramos para contrastar la efectividad de la popularidad es el recomendador aleatorio, pues representa el acierto que se consigue cuando "no se hace nada", cuando no se tiene en cuenta ningún factor a la hora de recomendar.

Podemos pues razonar sobre la efectividad de la popularidad en distintas situaciones valorando la diferencia entre popularidad y el recomendador aleatorio en dichas situaciones. En el caso particular que estamos tratando en esta sección, las situaciones vienen caracterizadas por la distribución de popularidad de los datos de entrada δ cuyo principal factor de interés es el sesgo: cuanto de populares son unos ítems respecto a otros y cuanto de activos son unos usuarios frente a otros.

Cabe destacar que en el presente trabajo vamos a considerar únicamente el efecto del sesgo de la distribución de popularidad de los ítems – cuantos votos ha recibido cada ítem – y vamos a ignorar el sesgo de la distribución por usuarios – cuantos ítems ha votado cada usuario – dejando dicho estudio como trabajo futuro.

5.3.1 Comparación analítica

Como acabamos de ver, las fórmulas 7 y 8 predicen con buena exactitud el valor de la precisión observada para los recomendadores aleatorio y popularidad, respectivamente. Sin embargo, es complicado comparar ambas fórmulas y, en particular, analizar cómo influye en ellas la distribución de popularidad. Sin embargo, en la formulación de popularidad empleamos unas simplificaciones que también son válidas para el recomendador aleatorio: independencia del usuario y existencia de un único ránking para todos los usuarios (esta última lleva implícita la suposición de que la partición mantiene la distribución original de los datos). Así, en la ecuación 8 basta con asumir que el ránking R es el producido por dicho recomendador aleatorio.

Si bien las simplificaciones anteriores permiten formular ambos recomendadores con una misma expresión, la presencia del tamaño de la intersección $\left|i_1 \cap \ldots \cap i_k^{rel}\right|$ sigue haciendo poco viable la comparación. Por ello, nos remontamos a la fórmula 5 y realizamos una nueva asunción: la independencia entre votos de distintos ítems.

$$p(rate(i_k), rel(i_k), rate(i_1), \ldots, rate(i_{k-1})|R) \sim p(rate, rel|i_k, R) \prod_{j=1}^{k-1} p\left(rate\middle|i_j, R\right)$$

Esta última simplificación implica que haber votado unos ciertos ítems no aporta información sobre si se ha votado otro ítem distinto. Pese a que no es necesariamente cierta, especialmente si el sesgo de la distribución por usuario es grande, no es descabellado asumir que las posibles imprecisiones afectan de forma similar a ambos recomendadores y, por tanto, asumirla no afecta a la validez de las conclusiones que aquí presentamos.

Por último, cabe considerar que en el planteamiento que estamos desarrollando interesa analizar la influencia de la distribución de popularidad por separado – manifestada a través de la variable $rate$ – sin considerar los posibles sesgos de otras distribuciones. Por ello, asumimos una hipótesis de neutralidad en el resto de variables, en particular y en lo que afecta a la fórmula anterior, consideramos que la relevancia no influye ni depende de ninguna otra variable: $p(rel\,, \cdot \mid \cdot) = p(rel)p(\cdot \mid \cdot)$.

En base a todas las hipótesis anteriores, la fórmula 5 resulta de la siguiente forma:

$$\mathbb{E}[\bar{P}@1|R] = \frac{(1-\rho)}{\rho}\, p(rel) \sum_{k=1}^{n} \prod_{j=1}^{k} \rho\, p\left(rate\middle|i_j, R\right)$$

En esta situación, y estimando las probabilidades $p(rate, rel|i)$ y $p(rate|i)$ mediante la distribución de popularidad de los ítems, $|i|/m \sim p(rate|i)$ e $|i^{rel}|/m \sim p(rate, rel|i)$, se puede ver que las popularidades total y relevante producen el mismo ránking. Así, la popularidad total ordena por $|i| \propto p(rate|i)$ mientras que la popularidad relevante ordena por $|i^{rel}| \propto p(rate, rel|i) = p(rate|i)p(rel) \propto\ p(rate|i) \propto |i|$, es decir, también por $|i|$.

Utilizando las estimaciones anteriores la fórmula se puede expresar como:

$$\mathbb{E}[\bar{P}@1|R] = \frac{(1-\rho)}{\rho}\, p(rel) \sum_{k=1}^{n} \prod_{j=1}^{k} \rho\, \frac{\left|i_j\right|}{m} \tag{9}$$

A continuación vamos a demostrar que el valor máximo de la precisión descrita en la anterior ecuación se alcanza precisamente con el ránking producido por los dos tipos de popularidad. Así mismo, veremos que este valor es mayor cuanto más sesgada es la distribución de popularidad por ítem.

Para la demostración empleamos el hecho de que cualquier ránking puede generarse a partir del ránking producido por popularidad – ordenado por $|i|$ – mediante una combinación de transposiciones de ítem adyacentes i_l, i_{l+1} donde $|i_l| > |i_{l+1}|$ [Anexo 1: Lema1]. Conociendo esta afirmación, sólo necesitamos probar que, dado un ránking $i_1, \dots i_l, i_{l+1}, \dots, i_n$ con $|i_l| > |i_{l+1}|$, intercambiar los ítems i_l y i_{l+1} produce un ránking con una menor precisión observada.

De acuerdo con la fórmula 9, la precisión en ambos ránkings es la siguiente:

$$\mathbb{E}[\bar{P}@1|i_1, \dots i_l, i_{l+1}, \dots, i_n] = \frac{(1-\rho)}{\rho}\, p(rel)\left(C_1 + \rho\frac{|i_l|}{m}\mathrm{C}_2 + \frac{|i_{l+1}|}{m}\rho\frac{|i_l|}{m}\mathrm{C}_2 + \mathrm{C}_3\right)$$

$$\mathbb{E}[\bar{P}@1|i_1, \dots i_{l+1}, i_l, \dots, i_n] = \frac{(1-\rho)}{\rho}\, p(rel)\left(C_1 + \rho\frac{|i_{l+1}|}{m}\mathrm{C}_2 + \frac{|i_l|}{m}\rho\frac{|i_{l+1}|}{m}\mathrm{C}_2 + \mathrm{C}_3\right)$$

Donde C_1, C_2 y C_3 son constantes en el sentido en que no dependen de i_l o i_{l+1} y, por tanto, no cambian al intercambiar ambos ítems[9].

La diferencia de precisión entre ambos ránkings es por tanto

$$(1-\rho)\, p(rel)\frac{\mathrm{C}_2}{m}(|i_l| - |i_{l+1}|) \geq 0$$

Es decir, el ránking original tiene una precisión observada mayor, tal y como queríamos demostrar. Además, la diferencia entre ambos ránkings es proporcional a $|i_l| - |i_{l+1}|$, esto es, a la diferencia de popularidad de los ítems i_l y i_{l+1}. Por tanto, cuanto más sesgada sea la distribución de popularidad mayor será esta diferencia entre ambos ránkings.

En base a esta demostración, se tiene que el ránking producido por popularidad tiene una precisión observada mayor que la de cualquier otro – en particular el producido por el recomendador aleatorio – y que su diferencia con cualquier otro ránking será mayor cuanto mayor sea el sesgo de la distribución de popularidad. También desde un punto de vista intuitivo se puede deducir que cuanto menor sesgo haya, es decir, cuanto más se parezcan unos ítems a otros, más se parecerán los ránkings de los recomendadores aleatorio y popularidad. Así, en una situación extrema en la que todos los ítems han recibido el mismo número de votos ambos recomendadores son equivalentes, pues eligen los ítems de forma uniforme y, por tanto, su precisión es la misma. En dicho caso la diferencia de precisión entre ambos ránkings valdría 0.

Cabe destacar que esta comparación – al igual que las que realizamos en el resto del documento – únicamente afecta a aquellos recomendadores y situaciones que cumplen con las hipótesis formuladas, en particular el hecho de que el usuario no es influyente. Así, estas conclusiones no afectan a recomendadores personalizados los cuales generalmente suelen presentar una precisión mayor que popularidad.

[9] Los valores de las tres constantes son:

$$C_1 = \sum_{k=1}^{l-1}\prod_{j=1}^{k}\rho\, p(rate|i_j, R)\;\; C_2 = \prod_{j=1}^{l-1}\rho\, p(rate|i_j, R)\;\; C_3 = \sum_{k=l+1}^{n}\prod_{j=1}^{k}\rho\, p(rate|i_j, R)$$

5.3.2 Comparación empírica

Para comprobar las conclusiones obtenidas en el desarrollo analítico, realizamos una simulación en la que generamos distintas distribuciones de ratings variando el sesgo en la distribución de popularidad por ítem. Para ello, utilizamos los parámetros – número de usuarios, número de ítems, número de votos y número de votos relevantes – del conjunto de datos de MovieLens descrito en la sección 2.

Para generar los ratings consideramos dos distribuciones de probabilidad, una por usuarios y otra por ítems que indican la probabilidad de que un cierto usuario/ítem sea el que realice/reciba el siguiente voto. De esta forma, la probabilidad de que un cierto usuario vote un determinado ítem se calcula como el producto de las dos distribuciones anteriores. En función de esta última probabilidad seleccionamos pares usuario-ítem sin reemplazo. Cuando un par es seleccionado asignamos un rating a dicho par. La relevancia del voto se decide de manera aleatoria con una cierta probabilidad establecida – $p(rel)$ – que estimamos a partir del número de votos relevantes. Así, la relevancia es independiente del resto de variables, tal y como asumíamos en las hipótesis del planteamiento formal.

Para las distribuciones de probabilidad de ítems y usuarios hemos empleado distribuciones de Pareto (o power law) cuyo valor para el usuario o ítem k-ésimo viene dado por la fórmula:

$$f(k) = C \cdot k^{-\alpha}$$

La constante C toma el valor adecuado para que al sumar sobre todos los ítems (o usuarios) se obtenga el valor 1. Respecto al exponente α, un valor mayor implica un mayor sesgo en la distribución, tal y como se puede observar en la Figura 11 donde se muestran distribuciones de este tipo para distintos exponentes.

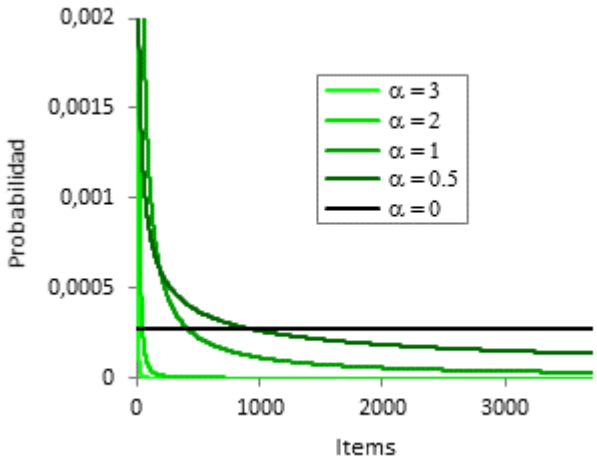

Figura 11. Distribuciones de Pareto normalizadas (la suma sobre el eje x es 1) correspondientes a distintos valores del exponente α. Cada distribución representa la probabilidad de que el siguiente voto sea asignado a cada ítem. En el eje *x* se indican 3706 ítems de acuerdo al número de ítems en el dataset de MovieLens. El eje *y* se muestra en los valores cercanos al 0 para poder apreciar la diferencia entre las distintas curvas.

En la simulación, la distribución de probabilidad por usuarios es siempre la misma y sigue una distribución de Pareto de exponente 3. Respecto a la distribución por ítems, hemos variado el exponente entre 0 y 3 para generar distribuciones con distinto sesgo y así poder comprobar cómo influye.

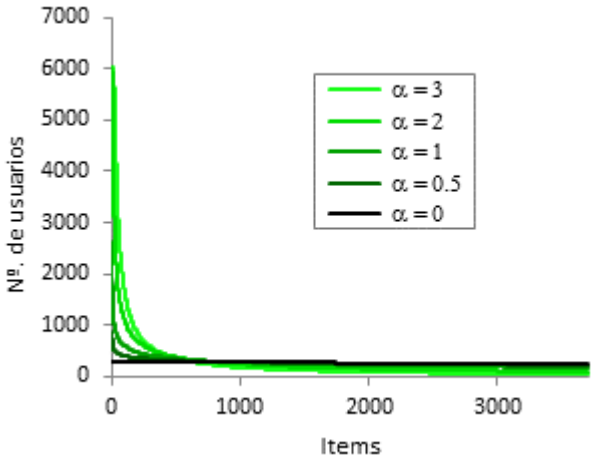

Figura 12. Distribuciones de popularidad por ítem (número de votos que ha recibido cada ítem) obtenidas en la simulación a partir de las distribuciones de probabilidad de la Figura 11.

En la Figura 12 se muestran las distribuciones de popularidad por ítem obtenidas tras simular la generación de ratings de la forma descrita anteriormente a partir de las distribuciones de probabilidad mostradas en la Figura 11 anterior. Cabe destacar la diferencia entre ambas gráficas: en la primera se muestra la probabilidad de que un ítem reciba el siguiente voto mientras que en la segunda se indican los votos que realmente ha recibido tras el proceso de generación de ratings en el que se emplea la probabilidad anterior. El hecho de que el muestreo de ítems y usuarios es sin reemplazo acota el máximo valor de popularidad que puede alcanzar cada ítem – valor que coincide con el número de usuarios totales – y conlleva la aparición de empates en los ítems más populares cuando el sesgo de la distribución de popularidad correspondiente es alto.

Una vez generada la distribución de ratings ejecutamos y evaluamos los recomendadores aleatorio y popularidad – total y relevante – mediante una partición aleatoria de parámetro $\rho = 0.8$. En esta situación, la evolución de la precisión para los tres recomendadores en función del sesgo de la distribución por ítem – definido por el exponente de la distribución de Pareto – se muestra en la Figura 13.

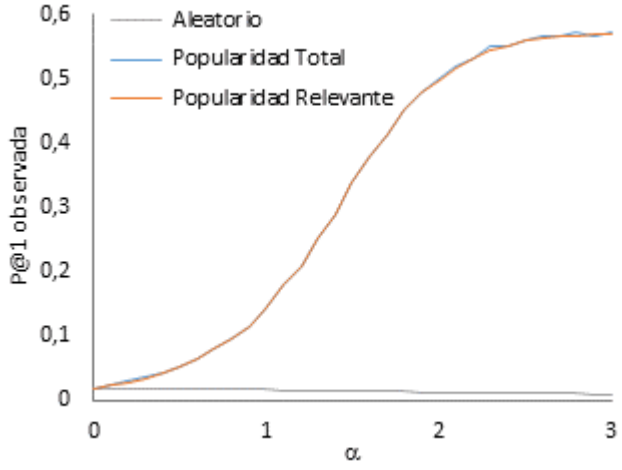

Figura 13. Evolución de la precisión observada de los recomendadores – aleatorio, popularidad total y popularidad relevante – en función del sesgo de la distribución de popularidad por ítem, determinado por el exponente α de la distribución de probabilidad.

Observamos en primer lugar que la precisión de las dos popularidades coincide, tal y como esperábamos. Respecto al recomendador aleatorio, su precisión prácticamente no varía en función del sesgo, sin embargo la precisión de la popularidad aumenta y con ella la diferencia con entre ambos recomendadores, tal y como predecíamos en el desarrollo analítico.

6. Influencia del descubrimiento

En el capítulo anterior hemos desarrollado un análisis en el que las distintas situaciones o entornos de recomendación vienen descritos por una distribución concreta de ratings δ. Sin embargo, en muchas ocasiones resulta deseable caracterizar la situación en función de la relación entre las distintas variables que intervienen en la generación de ratings; en concreto interesa considerar la relación de las variables rel y $rate$ con la forma en que se descubren los ítems en el entorno, es decir, con la variable $seen$. De esta forma es posible abordar cuestiones como el efecto que tiene publicitar los ítems más populares frente a hacerlo con los que no lo son tanto, o estudiar si una situación en la que únicamente se descubre lo no relevante podría perjudicar a la recomendación por popularidad hasta el punto de que fuese peor que la recomendación aleatoria.

De acuerdo con este planteamiento, cada situación viene caracterizada por una serie de asunciones acerca de las distribuciones de las distintas variables. En particular, es posible considerar la distribución de relevancia real – desconocida en el caso anterior porque únicamente fijabamos la distribución de ratings – lo que permite a su vez estudiar la precisión real. En esta nueva situación, interesa estudiar si la relación de la relevancia real con el descubrimiento o la votación puede causar contradicciones entre la precisión observada y la precisión real.

A la hora de realizar la formulación de las precisiones, cabe recordar que para medir la efectividad de la popularidad seguimos interesados en compararla con el recomendador aleatorio por lo que mantenemos las hipótesis de la sección 5.3.1 que se asumían para poder comparar ambos recomendadores: independencia del usuario, mantenimiento de la distribución original por parte de la partición, existencia de un único ránking que se presenta a todos los usuarios e independencia de la variable $rate$ en distintos ítems. Sin embargo, en este caso no realizamos ninguna asunción acerca de la dependencia entre la relevancia y el resto de variables pues consideramos que esta posible dependencia es conocida.

Bajo las hipótesis anteriores y remitiéndonos nuevamente al caso de una partición aleatoria las ecuaciones 3 y 4, para la esperanza de las precisiones observada y real, resultan de la siguiente forma:

$$\mathbb{E}[\bar{P}@1|R] = \sum_{k=1}^{n} (1-\rho) p(rate|rel, i_k, R)\, p(rel|i_k, R) \prod_{j=1}^{k-1} \rho\, p\big(rate\big|i_j, R\big) \qquad (10)$$

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} \big(1-\rho\, p(rate|rel, i_k, R)\big)\, p(rel|i_k, R) \prod_{j=1}^{k-1} \rho\, p\big(rate\big|i_j, R\big) \qquad (11)$$

De las ecuaciones anteriores deducimos [Anexo 1: Lema 2] que el ránking con mayor precisión observada es aquel que ordena los ítems según el siguiente valor:

$$\frac{p(rate|\, rel, i)\, p(rel|i)}{1-\rho\, p(rate|\, i)} \qquad (12)$$

Mientras que para obtener la máxima precisión real hay que ordenarlos en función del siguiente cociente.

$$\frac{p(rel|i)\big(1-\rho\, p(rate|\, rel, i)\big)}{1-\rho\, p(rate|\, i)} \tag{1}$$

De acuerdo con el estudio que pretendemos realizar, es deseable introducir la variable $seen$ en la formulación anterior para poder analizar su influencia en el resultado. Para ello, retomamos la situación descrita en la red bayesiana de la Figura 6 en la que se considera la conexión de esta variable con el resto de variables. Tal y como se observa en dicha figura, la variable $seen$ influye de manera directa en la votación, pues no es posible realizar un voto sin haber descubierto el ítem antes. Probabilísticamente esta última afirmación se formula de la siguiente forma:

$$p(rate|\neg seen) = 0$$

Por ello, un valor afirmativo en la variable $rate$ implica necesariamente que $seen$ también se cumple, de donde se puede deducir la siguiente descomposición:

$$p(rate|\,\cdot) = p(rate, seen|\,\cdot) = p(rate|seen,\cdot)\; p(seen|\,\cdot)$$

En este punto cabe señalar que consideramos que la decisión del usuario sobre si vota o no un ítem – que conoce – únicamente depende de si el ítem le gusta o no, no del ítem en concreto:

$$p(rate|seen, rel, i) \sim p(rate|seen, rel)$$
$$p(rate|seen, \neg rel, i) \sim p(rate|seen, \neg rel)$$

Introduciendo la variable $seen$ y empleando las reducciones anteriores las ecuaciones 10 y 11 presentan la siguiente forma:

$$\mathbb{E}[\bar{P}@1|R] = (1-\rho)\; p(rate|seen, rel) \sum_{k=1}^{n} p(seen|rel, i_k, R) p(rel|i_k, R) \cdot \prod_{j=1}^{k-1} \rho\; p\big(rate\big|seen, i_j, R\big) p\big(seen\big|\, i_j, R\big)$$

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} \big(1-\rho\; p(rate|seen, rel)\; p(seen|rel, i_k, R)\big)\; p(rel|i_k, R) \cdot \prod_{j=1}^{k-1} \rho\; p\big(rate\big|seen, i_j, R\big) p\big(seen\big|\, i_j, R\big)$$

Para deducir los ránkings óptimos introducimos la variable $seen$ en las expresiones 12 y 13 de los valores por los que hay que ordenar para obtener la máxima precisión observada y real, respectivamente. De esta forma, y tras eliminar las constantes que no dependen de los ítems y que aparecen multiplicando, dichos valores resultan en los siguientes dos cocientes – siendo $\bar{C}(i)$ el correspondiente a la precisión observada y $C(i)$ el de la precisión real.

$$\bar{C}(i) = \frac{p(seen|\, rel, i)\; p(rel|i)}{1-\rho\; p(rate|seen, i)\; p(seen|\, i)}$$

$$C(i) = \frac{p(rel|i)\big(1-\rho\; p(rate|seen, rel)\; p(seen|\, rel, i)\big)}{1-\rho\; p(rate|seen, i)\; p(seen|\, i)}$$

A partir de estas expresiones, a continuación comparamos las precisiones del recomendador aleatorio, popularidad total y popularidad relevante en función del comportamiento de las

variables $rate$ y $seen$, en particular de su dependencia con la relevancia. En primer lugar estudiamos la influencia del comportamiento del usuario, esto es, la influencia de votar más lo relevante o lo no relevante. Posteriormente analizamos cómo afecta que se descubran más unos ítems que otros, estudiando diversas situaciones como que los más descubiertos sean los más relevantes – o los menos – o que dicho descubrimiento sea independiente de la relevancia.

6.1 Comportamiento del usuario (rating)

Como ya adelantábamos anteriormente, consideramos que el comportamiento del usuario a la hora de decidir si realiza un voto o no – sobre un ítem que ha descubierto – únicamente depende de la relevancia. Por ello, caracterizamos este comportamiento mediante dos parámetros: $p(rate|seen, rel)$ y $p(rate|seen, \neg rel)$ que representan la tendencia de los usuarios a votar lo que les gusta y lo que no les gusta, respectivamente. En función del valor de estos parámetros tendremos unas situaciones u otras.

Para analizar dichas situaciones nos interesa, por tanto, estudiar cómo influyen los parámetros anteriores en la formulación de las precisiones. Para ello, consideramos un comportamiento neutral en las variables que no estamos analizando. En particular, consideramos que el descubrimiento no influye, que es uniforme e independiente del resto de variables.

$$p(seen|rel, i) = p(seen)$$

Con estas hipótesis ambas precisiones son proporcionales – y con ellas los cocientes $\bar{C}(i)$ y $C(i)$ – por lo que sus ránkings óptimos son realmente el mismo y se consiguen ordenando por siguiente factor[10].

$$\frac{p(rel|i)}{1 - \rho\, p(rate|seen, i)\, p(seen)}$$

Descomponiendo el valor de $p(rate|seen, i)$ vemos que el cociente anterior es una función creciente de la probabilidad de relevancia[11], es decir, que el ránking óptimo – el que produce el valor más alto en ambas precisiones – es aquel que está ordenado por relevancia. Consecuentemente, ordenar de forma contraria a la relevancia produce el peor ránking.

En base a estas afirmaciones analizamos cómo se comportan cada uno de los recomendadores de popularidad.

---

[10] El factor se calcula a partir de cualquiera de los cocientes $\bar{C}(i)$ y $C(i)$ al eliminar la multiplicación por las constantes que no dependen del ítem.

[11] La descomposición de $p(rate|seen, i)$ es

$$p(rate|seen, i) = \big(p(rate|seen, rel) - p(rate|seen, \neg rel)\big)\, p(rel|i) + p(rate|seen, \neg rel)$$

Por lo que se cumple que

$$\frac{p(rel|i)}{1 - \rho\, p(rate|seen, i)\, p(seen)} = \frac{p(rel|i)}{a\, p(rel|i) + b}$$

Donde $a = \rho\, p(seen)\big(p(rate|seen, rel) - p(rate|seen, \neg rel)\big)$ y $b = 1 - \rho\, p(seen) p(rate|seen, \neg rel)$.
El signo de la derivada de la función $f(x) = \frac{x}{a\,x + b}$ es el signo de b, que en este caso es positivo porque ρ, $p(seen)$ y $p(rate|seen, \neg rel)$ son probabilidades y, por tanto, menores que 1.
De esta forma se deduce que la función es creciente con $p(rel|i)$.

Popularidad relevante

Popularidad relevante ordena por el número de votos relevantes, por lo que es equivalente a ordenar por la probabilidad $p(rate, rel|i)$ y, en la situación que estamos tratando, se cumple que $p(rate, rel|i)$ es proporcional a la relevancia.

$$p(rat, rel|i) = p(rate|rel, i)\, p(rel|i) = p(rate|seen, rel, i)\, p(seen|rel, i)\, p(rel|i)$$
$$= p(rating|seen, rel)\, p(seen)\, p(rel|i) = c \cdot p(rel|i)$$

Consecuentemente, popularidad relevante también ordena por relevancia y, por tanto, el ránking que produce es óptimo en ambas precisiones, sean cuales sean los valores de los parámetros $p(rate|seen, rel)$ y $p(rate|seen, \neg rel)$. Así pues tenemos[12]:

$$\mathbb{E}[\bar{P}@1|rpop] \geq \mathbb{E}[\bar{P}@1|rnd], \mathbb{E}[\bar{P}@1|pop]$$
$$\mathbb{E}[P@1|rpop] \geq \mathbb{E}[P@1|rnd], \mathbb{E}[P@1|pop]$$

Popularidad total

Siguiendo el mismo razonamiento que para la popularidad relevante, se tiene que la popularidad total ordena por $p(rate|i)$. Introduciendo las variables $seen$ y rel esta probabilidad se puede descomponer de la siguiente forma:

$$p(rate|i) = \Big(\big(p(rate|seen, rel) - p(rate|seen, \neg rel)\big)\, p(rel|i)$$
$$+ p(rate|seen, \neg rel)\Big)\, p(seen)$$

Es decir, en función de los valores de los parámetros $p(rate|seen, rel)$ y $p(rate|seen, \neg rel)$ popularidad relevante ordenará de una forma u otra con respecto a la relevancia. Distinguimos tres situaciones:

- $p(rate|seen, rel) = p(rate|seen, \neg rel)$: Esta igualdad representan una situación en la que el comportamiento del usuario no depende de la relevancia ya que vota indistintamente los ítems que le gustan y los que no. En este caso, popularidad total es equivalente al recomendador aleatorio porque $p(rate|i)$ es constante, es decir, todos los ítems tienen la misma popularidad y la elección entre quien se recomienda antes y quién después es aleatoria.

$$\mathbb{E}[\bar{P}@1|rpop] \geq \mathbb{E}[\bar{P}@1|rnd] = \mathbb{E}[\bar{P}@1|pop]$$
$$\mathbb{E}[P@1|rpop] \geq \mathbb{E}[P@1|rnd] = \mathbb{E}[P@1|pop]$$

- $p(rate|seen, rel) > p(rate|seen, \neg rel)$: Los usuarios votan con más frecuencia los ítem que les gustan que los que no. En esta situación popularidad total ordena por relevancia, por lo que produce el mismo ránking que popularidad relevante. Como ya hemos dicho, este ránking es óptimo para ambas precisiones por lo que, particularmente, es mejor que el ránking promedio del recomendador aleatorio:

$$\mathbb{E}[\bar{P}@1|rpop] = \mathbb{E}[\bar{P}@1|pop] > \mathbb{E}[\bar{P}@1|rnd]$$
$$\mathbb{E}[P@1|rpop] = \mathbb{E}[P@1|pop] > \mathbb{E}[P@1|rnd]$$

[12] Abuso de notación: se asume que los tres recomendadores producen el mismo ránking para todos los usuarios, por lo que se puede identificar el ránking con el recomendador y substituir $\mathbb{E}[P@1|R\ generado\ por\ \theta]$ por $\mathbb{E}[P@1|\theta]$. En este caso rnd hace referencia al recomendador aleatorio y pop y $rpop$ a los recomendadores por popularidad total y relevante, respectivamente.

- $p(rate|seen, rel) < p(rate|seen, \neg rel)$: Los usuarios votan con más frecuencia los ítems que no les gustan que los que sí lo hacen. En esta situación, popularidad total ordena de forma contraria a la relevancia por lo que el ránking que produce es el peor posible:

$$\mathbb{E}[\bar{P}@1|rpop] \geq \mathbb{E}[\bar{P}@1|rnd] \geq \mathbb{E}[\bar{P}@1|pop]$$
$$\mathbb{E}[P@1|rpop] \geq \mathbb{E}[P@1|rnd] \geq \mathbb{E}[P@1|pop]$$

En resumen, observamos que para reducir la eficacia de la popularidad relevante no basta con alterar el comportamiento de los usuarios hacia actitudes poco frecuentes – como votar más lo no relevante – ya que este algoritmo consigue presentar recomendaciones con una precisión superior al recomendador aleatorio en todas las situaciones. Por el contrario, popularidad total sí que puede ver afectada su eficacia si los usuarios tienden a votar lo que no les gusta, pues los ítems con más votos serán precisamente los menos relevantes. Afortunadamente, en la mayoría de conjuntos de datos públicos hay más votos positivos que negativos, por lo que se está en la segunda situación – asumiendo un descubrimiento uniforme – y ambas popularidades son equivalentes y superiores al recomendador aleatorio.

6.2 Descubrimiento

Para estudiar los efectos específicos de la distribución del descubrimiento en las precisiones real y observada anulamos la influencia de las variables que no interesan. Concretamente, consideramos que la variable $rate$, cuya influencia ya hemos estudiado, es independiente de la relevancia.

$$p(rate|seen, rel) = p(rate|seen, \neg rel) = p(rate|seen)$$

Se cumple, por tanto, que $p(rate|seen, i)$ es constante respecto al ítem, por lo que la vamos a denotar con la letra C. Con esta nueva notación, los cocientes $\bar{C}(i)$ y $C(i)$ que determinan el orden óptimo resultan de la siguiente forma:

$$\bar{C}(i) = \frac{p(seen|rel, i)\, p(rel|i)}{1 - \rho\, C\, p(seen|\, i)}$$
$$C(i) = \frac{p(rel|i)\big(1 - \rho\, C\, p(seen|\, rel, i)\big)}{1 - \rho\, C\, p(seen|\, i)}$$

En esta situación, el principal aspecto que interesa estudiar de la distribución de descubrimiento es su posible relación con la relevancia, relación que se resume mediante la expresión $p(seen|rel, i)$. En este sentido cabe diferenciar dos casos extremos: que el descubrimiento sea independiente del ítem conocida su relevancia ($p(seen|rel, i) = p(seen|rel)$) o que sea independiente de la relevancia conocido el ítem ($p(seen|rel, i) = p(seen|i)$). A continuación estudiaremos los comportamientos que pueden producirse en ambas situaciones. Sin embargo, cabe señalar que en las situaciones típicas no se suelen encontrar fenómenos de independencia, por lo que tiene sentido considerar también el caso en que el descubrimiento depende tanto de la relevancia como del ítem.

6.2.1 Descubrimiento independiente del ítem dada la relevancia

La dependencia entre el descubrimiento y la relevancia representa la habilidad del usuario – junto con los buscadores, recomendadores, etc. – para descubrir lo que le gusta. Por ello, la independencia entre descubrimiento e ítem representa una situación donde esta habilidad es el

único factor que determina lo que se descubre o no. No se considera, por tanto, ninguna otra característica del ítem, como lo conocido que ya sea o lo interesante que resulte para alguna compañía el publicitarlo. En esta situación, todos los ítems relevantes tenderán a ser conocidos en la misma medida y lo mismo ocurrirá con los no relevantes. Formalmente, esta asunción de independencia se formula de la siguiente forma:

$$p(seen|rel, i) = p(seen|rel)$$
$$p(seen|\neg rel, i) = p(seen|\neg rel)$$

Bajo este planteamiento, las distintas situaciones posibles vienen caracterizadas por los valores de $p(seen|rel)$ y de $p(seen|\neg rel)$, es decir, por la dependencia entre descubrimiento y relevancia, por si se descubre más lo relevante que lo no relevante o al contrario. El estudio de estas situaciones sigue una estructura muy similar al análisis de la influencia del comportamiento del usuario.

Introduciendo la hipótesis de independencia del ítem dada la relevancia, las precisiones vuelven a ser proporcionales y su orden óptimo viene determinado por la expresión[13]:

$$\frac{p(rel|i)}{1 - \rho\ C\ p(seen|\ i)}$$

Este cociente es una función creciente de la relevancia, tal y como se puede ver descomponiendo la probabilidad $p(seen|\ i)$[14], por lo que nuevamente el ránking óptimo es aquel que está ordenado por relevancia. En esta situación analizamos los recomendadores de popularidad por separado para estudiar el ránking que producen.

Popularidad relevante

Como ya hemos explicado anteriormente, popularidad relevante ordena por $p(rate, rel|i)$, que vuelve a ser proporcional a la relevancia.

$$p(rate, rel|i) = p(rate|seen, rel) \cdot\ p(seen|rel) \cdot\ p(rel|i) = c \cdot p(rel|i)$$

Es decir, nuevamente popularidad relevante ordena los ítems del ránking por relevancia y, por tanto, dicho ránking es óptimo.

$$\mathbb{E}[\bar{P}@1|rpop] \geq\ \mathbb{E}[\bar{P}@1|rnd], \mathbb{E}[\bar{P}@1|pop]$$
$$\mathbb{E}[P@1|rpop] \geq\ \mathbb{E}[P@1|rnd], \mathbb{E}[P@1|pop]$$

Popularidad total

Respecto a popularidad total, ordena por $p(rate|i)$ que, en estas circunstancias, es proporcional al descubrimiento.

$$p(rate|i) = p(rate|seen, i) \cdot p(seen|i) = p(rate|seen) \cdot p(seen|i) = c \cdot p(seen|i)$$

---

[13] Como en la situación anterior, este factor se calcula a partir de los cocientes $\bar{C}(i)$ y $C(i)$ eliminando la multiplicación por las constantes que no dependen del ítem.

[14] Descomponiendo $p(seen|\ i) = \big(p(seen|rel) - p(seen|\neg rel)\big)\ p(rel|i) + p(seen|\neg rel)$ se obtiene que

$$\frac{p(rel|i)}{1 - \rho\ C\ p(seen|\ i)} = \frac{p(rel|i)}{a\ p(rel|\ i) + b}$$

Con b positivo. Por lo que, siguiendo un razonamiento análogo a la nota 11, se deduce que el cociente es creciente con $p(rel|i)$.

Dicho descubrimiento se puede descomponer de la siguiente forma en función de los parámetros $p(seen|rel)$ y $p(seen|\neg rel)$.

$$\begin{aligned} p(seen|i) &= p(seen|rel)p(rel|i) + p(seen|\neg rel)\big(1 - p(rel|i)\big) \\ &= \big(p(seen|rel) - p(seen|\neg rel)\big)p(rel|i) + p(seen|\neg rel) \end{aligned}$$

La situación es la misma que en el caso anterior, ya que en función de la comparativa entre $p(seen|rel)$ y $p(seen|\neg rel)$, popularidad total ordenará de una forma u otra con respecto a la relevancia. Se distinguen tres casos:

- $p(seen|rel) = p(seen|\neg rel)$: El descubrimiento no depende de la relevancia y, dado que tampoco depende del ítem, es independiente de todas las variables. Esta situación es la misma que la planteada en el primer punto de la sección 6.1 en el que se estudia la influencia del comportamiento del usuario. Así, en este caso asumimos un descubrimiento y una votación independientes del resto de variables, por lo que se cumple que $p(rate|i)$ es constante y, con ello, que popularidad total es equivalente al recomendador aleatorio.

$$\begin{aligned} \mathbb{E}[\bar{P}@1|rpop] \geq\ & \mathbb{E}[\bar{P}@1|rnd] = \mathbb{E}[\bar{P}@1|pop] \\ \mathbb{E}[P@1|rpop] \geq\ & \mathbb{E}[P@1|rnd] = \mathbb{E}[P@1|pop] \end{aligned}$$

- $p(seen|rel) > p(seen|\neg rel)$: Se descubren más los ítems relevantes que los que no lo son. En esta situación, popularidad total también ordena por relevancia – genera el ránking óptimo – y se cumplen las siguientes desigualdades:

$$\begin{aligned} \mathbb{E}[\bar{P}@1|rpop] &= \mathbb{E}[\bar{P}@1|pop] > \mathbb{E}[\bar{P}@1|rnd] \\ \mathbb{E}[P@1|rpop] &= \mathbb{E}[P@1|pop] > \mathbb{E}[P@1|rnd] \end{aligned}$$

- $p(seen|rel) \leq p(seen|\neg rel)$: Se descubren más los ítems que no gustan que los que sí lo hacen. En esta situación popularidad total ordena de forma contraria a la relevancia por lo que el ránking es el peor posible:

$$\begin{aligned} \mathbb{E}[\bar{P}@1|rpop] \geq\ & \mathbb{E}[\bar{P}@1|rnd] \geq \mathbb{E}[\bar{P}@1|pop] \\ \mathbb{E}[P@1|rpop] \geq\ & \mathbb{E}[P@1|rnd] \geq \mathbb{E}[P@1|pop] \end{aligned}$$

Nuevamente la eficacia de la popularidad relevante se mantiene en todas las situaciones, incluso aunque el descubrimiento sea contrario a la relevancia. Respecto a la popularidad total, si se vota indistintamente de la relevancia pero se descubre más lo negativo – tercer caso – el efecto es el mismo que si se votará más lo negativo. Así, los ítems menos relevantes se descubren más por lo que acaparan más ratings y son los que popularidad total acaba recomendando. De forma análoga se explican los resultados para el segundo caso, en el que descubrimiento y relevancia presentan una dependencia positiva.

6.2.2 Descubrimiento independiente de la relevancia dado el ítem

Hasta el momento, en todas las situaciones planteadas popularidad relevante es siempre más eficaz que el recomendador aleatorio. Respecto a popularidad total, dichas situaciones las resumimos en dos casos: los ítems más votados son los más relevantes (popularidad total es óptimo) o los menos relevantes (popularidad total es peor que el recomendador aleatorio).

Resulta intuitivo, sin embargo, pensar que invirtiendo las distribuciones de relevancia y descubrimiento popularidad relevante debería ver afectada en algunos casos su eficacia. Hasta ahora, hemos considerado una dependencia negativa entre $seen$ y rel – la misma para todos los ítems – sin embargo, estas son variables binarias que se evalúan para cada par usuario-ítem.

Cuando pensamos que sus distribuciones deberían ser contrarias nos estamos refiriendo realmente a la distribución del número de usuarios a los que les gusta /conocen un cierto ítem, es decir, queremos dar a entender que cuando un ítem le gusta a muchos usuarios es descubierto por pocos. Esta afirmación hace referencia a la relación entre las distribuciones $p(seen|i)$ y $p(rel|i)$, no entre las variables binarias $seen$ y rel. Es posible, por tanto, realizar una serie de hipótesis sobre la relación entre las variables y otras sobre sus distribuciones de probabilidad sobre los ítems.

Para simplificar estudiaremos dos casos extremos: que $p(seen|i)$ y $p(rel|i)$ mantengan el mismo orden en los ítems y que generen ordenes contrarios. Entre ambos extremos se encuentran una gran cantidad de situaciones que resulta inviable analizar pero que intuitivamente se pueden interpretar en función de lo cerca que estén de un extremo u otro.

Para poder realizar este estudio vamos a asumir que el descubrimiento es independiente de la relevancia dado el ítem.

$$p(seen|rel, i) = p(seen|i)$$

La dependencia entre el descubrimiento y el ítem refleja el esfuerzo de los diferentes ítems – o más concretamente de aquellos que los crean, comercializan, publicitan etc. – por darse a conocer a (el mayor número de) los usuarios. Independencia de la relevancia significa que este esfuerzo es altamente desigual entre ítems, hasta el punto de que la diferencia de relevancia (calidad, utilidad, etc.) apenas juega un papel perceptible

Como ya explicamos anteriormente, y aunque resulte poco intuitivo, es posible asumir esta independencia entre ambas variables pero posteriormente considerar que sus distribuciones de probabilidad presentan una cierta relación – generan el mismo orden o el contrario, por ejemplo.

Bajo esta hipótesis, los cocientes $\bar{C}(i)$ y $C(i)$ se simplifican de la siguiente forma:

$$\bar{C}(i) = \frac{p(rel|i)\ p(seen|i)}{1 - \rho\ C\ p(seen|\ i)}$$
$$C(i) = p(rel|i)$$

Es decir, que en esta situación, ordenar por relevancia produce un ránking óptimo pero sólo en cuanto a la precisión real. Respecto a la precisión observada no se pueden realizar asunciones tan generales, pero se cumple que si $p(seen|i)$ y $p(rel|i)$ mantienen el mismo orden entonces ordenar por relevancia también obtiene el mejor resultado.

Para analizar los ránkings que producen los recomendadores de popularidad descomponemos las probabilidades $p(rate, rel|i)$ y $p(rate|i)$ de la siguiente forma para que queden en función de $p(seen|i)$ y $p(rel|i)$.

$$p(rate, rel|i) = p(rate|seen, rel) \cdot\ p(seen|i) \cdot p(rel|i) = c \cdot p(seen|i) \cdot p(rel|i)$$

$$p(rate|i) = p(rate|seen) \cdot p(seen|i) = c \cdot p(seen|i)$$

Se cumple, por tanto que, popularidad relevante ordena de acuerdo al producto de $p(seen|i)$ y $p(rel|i)$, mientras que popularidad total ordena únicamente por $p(seen|i)$.

En base a las anteriores consideraciones, veamos la comparativa entre los recomendadores en función de la relación entre $p(seen|\ \ )$ y $p(rel|i)$.

$p(seen|i)$ y $p(rel|i)$ generan el mismo orden

Si $p(seen|i)$ mantiene el mismo orden que $p(rel|i)$, entonces tanto popularidad relevante como popularidad total ordenan por relevancia y, por tanto, producen ránkings óptimos en ambas precisiones[15].

$$\mathbb{E}[P@1|pop] = \mathbb{E}[P@1|rpop] \geq \mathbb{E}[P@1|rnd]$$
$$\mathbb{E}[\bar{P}@1|pop] = \mathbb{E}[\bar{P}@1|rpop] \geq \mathbb{E}[\bar{P}@1|rnd]$$

$p(seen|i)$ y $p(rel|i)$ presentan un orden contrario

En este caso popularidad total ordena de forma contraria a la relevancia, por lo que respecto a la precisión real produce el peor ránking.

$$\mathbb{E}[P@1|pop] \leq \mathbb{E}[P@1|rnd], \mathbb{E}[P@1|rpop]$$

Sin embargo, con respecto al resto de situaciones no es posible realizar ninguna afirmación, ya que todas son posibles. Así, en la Tabla 3 se muestran tres pequeños ejemplos en los cuales se producen distintas ordenaciones de los recomendadores, tanto en precisión real como en observada.

Cada ejemplo representa un entorno en el que únicamente existen dos ítems, denotados por las letras i y j. Cada ítem lleva asociado dos valores, la fracción de usuarios a los que les gusta – equivalente a $p(rel|i)$ y representado en la gráfica en color rosa claro – y la fracción de usuarios que conocen el ítem - $p(seen|i)$ en color verde claro.

En la parte superior de la Tabla 3 se muestra, para cada ejemplo, un diagrama de barras con los valores de las distribuciones anteriores de cada ítem. Observamos que en todos los casos dichas distribuciones producen ordenaciones opuestas: sin en un ítem la relevancia es mayor que en el otro, entonces el descubrimiento es menor

Debajo de los diagramas se indica el ránking que produce cada recomendador, junto con la precisión observada de dicho ránking. Para realizar estos cálculos hemos asumido una tasa de entrenamiento ρ de 0.8 y que $p(rating|\ seen, rel) = p(rating|\ seen) = 1$, es decir, que los usuarios votan todo lo que descubren con independencia de si les gusta o no. Dado que sólo existen dos ítems, únicamente son posibles dos ránkings: $< i, j >$ y $< j, i >$. En el caso del recomendador aleatorio no se indica el ránking porque puede ser cualquiera de los dos, y su precisión la calculamos como la media de las precisiones de los dos ránkings posibles.

En la parte inferior resumimos la ordenación de los recomendadores, tanto en precisión observada como en real. Para la precisión real no es necesario realizar los cálculos de forma explícita porque ya sabemos que el mejor ránking es aquel en el que los ítems están ordenados de acuerdo a su relevancia, es decir, de acuerdo al valor de $p(rel|i)$.

[15]Ya se comentó anteriormente que si $p(seen|i)$ y $p(rel|i)$ mantenían el mismo orden, ordenar por relevancia también producía un ránking óptimo en la precisión observada

	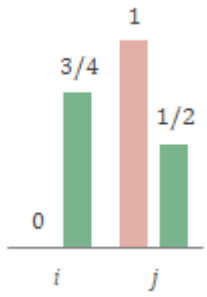		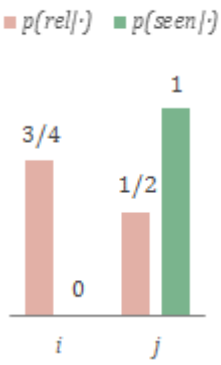		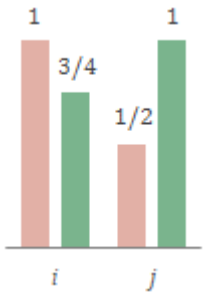	
	$\mathbb{E}[\bar{P}@1\|\theta]$	ránking	$\mathbb{E}[\bar{P}@1\|\theta]$	ránking	$\mathbb{E}[\bar{P}@1\|\theta]$	ránking
$rpop$	0.06	$< j, i >$	0.1	$< j, i >$	0.210	$< i, j >$
pop	0.1	$< i, j >$	0.1	$< j, i >$	0.220	$< j, i >$
rnd	0.08	--	0.05	--	0.215	--
	$\bar{P}@1 : rpop \geq rnd \geq pop$ $P@1 : rpop \geq rnd \geq pop$		$\bar{P}@1 : rpop = pop \geq rnd$ $P@1 : rpop = pop \leq rnd$		$\bar{P}@1 : pop \geq rnd \geq rpop$ $P@1 : pop \leq rnd \leq rpp$	

Tabla 3. Ejemplos de situaciones en las que se producen distintas ordenaciones de los recomendadores, tanto en precisión observada como en real. Cada situación viene descrita por el diagrama de barras superior, que indica los valores de las distribuciones $p(rel|i)$ y $p(seen|i)$ de cada ítem. Para cada ejemplo se indica la precisión observada de cada recomendador junto con el ránking que produce. Dicha precisión observada se ha calculado asumiendo que $\rho = 0.8$ y que $p(rating|\ seen) = 1$. En la parte inferior se indica la ordenación de los recomendadores en cuanto a precisión observada y a precisión real.

A continuación analizamos y explicamos cada uno de los ejemplos.

- En el primer ejemplo, el ítem *i* no le gusta a ningún usuario, pero lo conocen tres cuartas partes del total de usuarios. Por el contrario, el ítem *j* le gusta a todo el mundo pero sólo lo conocen la mitad de los usuarios.

  Para comprobar que este ejemplo es consistente con las hipótesis que estamos trabajando en este planteamiento, esto es, que $seen$ y rel son independientes dado el ítem, observamos que se cumplen las siguientes igualdades:

  $$p(seen, rel|i) = 0 = p(seen|i) \cdot p(rel|i)$$
  $$p(seen, rel|j) = p(seen|j) = p(seen|j) \cdot p(rel|j)$$

  Aquí vemos claramente un ejemplo en el que las variables $seen$ y rel son independientes, pero $p(seen|i)$ y $p(rel|i)$ son distribuciones contrarias. De hecho, los tres ejemplos presentan situaciones en las que se cumple este comportamiento.

  Respecto a los ránkings generados por los recomendadores, popularidad total produce el ránking $< i, j >$ dado que se cumple la siguiente desigualdad:

  $$p(seen|i) > p(seen|j)$$

  Popularidad relevante, sin embargo, genera el ránking contrario $< j, i >$ ya que:

  $$p(seen, rel|j) = p(seen|j) > 0 = p(seen, rel|i)$$

La precisión observada de cada ránking se calcula a partir de la fórmula 10 de la siguiente forma:

$$\mathbb{E}[\bar{P}@1|< i,j >] = (1-\rho)\big(p(seen|i)\ p(rel|i) + \rho\ p(seen|j)\ p(rel|j)\ p(seen|i)\big)$$
$$= (1-\rho)\big(0 + \rho\ p(seen|j)\ p(seen|i)\big)$$

$$\mathbb{E}[\bar{P}@1|< j,i >] = (1-\rho)\big(p(seen|j)\ p(rel|j) + \rho\ p(seen|i)\ p(rel|i)\ p(seen|j)\big)$$
$$= (1-\rho)(p(seen|j) + 0)$$

En base a los cálculos anteriores se observa que el ránking $< j,i >$ es el ránking óptimo en cuanto a precisión observada, ya que se cumple la siguiente desigualdad:

$$(1-\rho)\ \rho\ p(seen|j)\ p(seen|i) < (1-\rho)\ p(seen|j)$$

Este mismo ránking también es óptimo en cuanto a precisión real porque el ítem j es más relevante que el ítem i, es decir, se cumple que $p(rel|j)$ es mayor que $p(rel|i)$.

Se tiene, por tanto, que popularidad relevante genera el mejor ránking, tanto en precisión observada como en real, mientras que popularidad total produce el peor. El recomendador aleatorio se encuentra entre ambos extremos, pues puede generar cualquiera de los dos ránkings con la misma probabilidad.

Cabe destacar que todas las aseveraciones realizadas en este ejemplo siguen siendo válidas si en lugar de fijar los valores de $p(seen|j)$ y $p(seen|i)$ asumimos simplemente que el segundo valor es mayor que el primero, es decir, que el ítem i es más conocido que el ítem j.

- El segundo ejemplo presenta una situación parecida al anterior pero intercambiando la relevancia y el descubrimiento. Así, el ítem i le gusta a tres cuartas partes de los usuarios, pero ninguno lo conoce. El ítem j, por su parte, le gusta sólo a la mitad pero es conocido por todo el mundo.

  En este caso, y realizando un razonamiento análogo al primer ejemplo, se obtiene que ambas popularidades presentan el miso ránking $< j,i >$. Este ránking es óptimo en cuento a precisión observada pero resulta el peor si medimos su precisión real. Se trata, por tanto, de un ejemplo en el que existe una contradicción entre ambos tipos de precisión. Si en un experimento real ocurriera esta contradicción, se estaría valorando como mejor un algoritmo que en realidad es peor.

  Nuevamente cabe mencionar que el razonamiento seguido en este ejemplo sigue siendo válido si se asume simplemente que la relevancia del ítem i es mayor que la del ítem j.

- El tercer ejemplo es ligeramente diferente y su objetivo es mostrar una situación en la que popularidad relevante presenta una precisión observada menor que los otros dos recomendadores.

  Así, en esta situación el ítem i gusta a todo el mundo y lo conocen tres cuartas partes de los usuarios. Por el contrario, el ítem j le gusta sólo a la mitad pero es conocido por todos.

  De forma análoga al primer ejemplo comprobamos que $seen$ y rel son independientes. De igual forma se puede ver que popularidad relevante genera el ránking $< i,j >$ y popularidad total el ránking $< j,i >$.

  Los cálculos de la precisión observada se muestran a continuación.

$$\mathbb{E}[\bar{P}@1|<i,j>] = (1-\rho)\big(p(seen|i)\,p(rel|i) + \rho\,p(seen|j)\,p(rel|j)\,p(seen|i)\big)$$
$$= (1-\rho)\big(p(seen|i) + \rho\,p(rel|j)\,p(seen|i)\big)$$

$$\mathbb{E}[\bar{P}@1|<j,i>] = (1-\rho)\big(p(seen|j)\,p(rel|j) + \rho\,p(seen|i)\,p(rel|i)\,p(seen|j)\big)$$
$$= (1-\rho)\big(p(rel|j) + \rho\,p(seen|i)\big)$$

En este caso, para que el ránking $<j,i>$ produzca el mejor resultado en cuanto a precisión observada es necesario que se cumpla la siguiente desigualdad:

$$p(seen|i) + \rho\,p(rel|j)\,p(seen|i) < p(rel|j) + \rho\,p(seen|i)$$

O, equivalentemente, que se cumpla:

$$p(rel|j) > (1-\rho)\frac{p(seen|i)}{\big(1-\rho\,p(seen|i)\big)}$$

Los datos concretos del ejemplo cumplen dicha desigualdad, pero cualquier otro par de valores que la cumpla seguiría validando la situación.

En resumen, los tres ejemplos anteriores muestran que cuando las distribuciones $p(seen|i)$ y $p(rel|i)$ son contrarias lo único que se puede saber con seguridad es que la precisión real de popularidad total es peor que la del resto de recomendadores. Respecto al resto de cuestiones existe incertidumbre: la precisión real de popularidad relevante puede ser mayor o menor que la del recomendador aleatorio (ejemplos 1 y 2), cualquier orden entre los tres recomendadores en cuanto a precisión observada es posible (ejemplos 1 y 3), y pueden existir contradicciones entre ambas precisiones (ejemplos 2 y 3).

6.2.3 Descubrimiento dependiente de la relevancia y el ítem

Las situaciones anteriores, pese a que nos permiten simplificar el estudio para visualizar y entender casos extremos, asumen unas ciertas hipótesis de independencia que no es frecuente encontrar en situaciones reales. En todas ellas, además, ordenar por relevancia (real) produce el mejor ránking con respecto a la precisión real, por lo que cabe preguntarse si esto es extensible a cualquier otra situación en la que no se produzcan las condiciones de independencia exigidas en los casos anteriores.

La respuesta es no. De hecho, recordamos que para obtener una precisión real óptima es necesario ordenar los ítems por el siguiente cociente:

$$C(i) = \frac{p(rel|i)\big(1-\rho\,C\,p(seen|\,rel,i)\big)}{1-\rho\,C\,p(seen|\,i)}$$

Es decir, que en ausencia de hipótesis de independencia los valores de ρ y C pueden ser determinantes a la hora de concluir si ordenar por relevancia produce el mejor ránking o no. De hecho, si ambos valores son muy cercanos a 1 el cociente anterior resulta en $p(rel|\neg seen, i)$. Es decir, cuando todo lo conocido se vota ($C=1$) y todo lo votado es asignado a entrenamiento ($\rho=1$), el mejor ránking – en cuanto a precisión real – es el que ordena por el número de opiniones relevantes dentro de las que no se conocen, pues dichas opiniones son precisamente con las que se va a evaluar.

La influencia de ρ no deja de ser sorprendente, pues implica que con un mismo conjunto de datos, emplear una tasa de entrenamiento u otra puede conllevar obtener una precisión real

óptima o no. Casualmente, en el experimento llevado a cabo para contrastar empíricamente la influencia del descubrimiento tiene lugar una de estas situaciones. Dicho experimento y sus resultados se explican en detalle a continuación, en la sección 6.3, pero en este punto cabe destacar que para que este comportamiento se produzca no es necesario que las distribuciones de relevancia y descubrimiento sean contrarias, como cabría esperar. Es decir, que incluso en un escenario en el que se descubre más lo relevante, con algunas tasas de entrenamiento es posible obtener una precisión real inferior al ordenar por relevancia real que al emplear un orden aleatorio.

En resumen, la dependencia del descubrimiento con la relevancia y el ítem plantea una situación en la que cualquier comportamiento de los recomendadores es posible.

6.3 Observación empírica

Para valorar empíricamente la influencia del descubrimiento en la evaluación offline no es viable emplear los conjuntos de datos típicamente utilizados para realizar dicha evaluación. La mayoría de estos conjuntos provienen de tiendas o aplicaciones online (MovieLens, Last.fm, Netflix…) en las que los usuarios voluntariamente manifiestan sus opiniones acerca de ítems que han descubierto previamente. Este descubrimiento se suele realizar de forma externa a la aplicación, por lo que es imposible conocer cómo se distribuye en aquellos ítems que los usuarios no han votado.

Tampoco se conoce la distribución completa de la relevancia real, sino únicamente la que reflejan los votos de los usuarios, la relevancia observada. Además, como ya introdujimos en la sección 2.2, estos conjuntos presentan distribuciones de popularidad fuertemente sesgadas que nos llevan a preguntarnos si esa relevancia que observamos – y que empleamos para recomendar y evaluar – es realmente una muestra representativa de la relevancia real o está influenciada por los posibles sesgos de la distribución de descubrimiento.

Por ello, interesa generar un conjunto de votos en el que, por un lado, la relevancia observada sea una muestra fiel de la distribución de relevancia real subyacente – para lo cual vamos a eliminar los sesgos de descubrimiento – y, por otro, se conozca la distribución de descubrimiento o pueda ser estimada – sin intervención de sesgo alguno – a partir de la muestra.

De acuerdo con este doble objetivo, hemos llevado a cabo un experimento con usuarios reales en el que, en ausencia de sesgos de descubrimiento, hemos obtenido sus preferencias acerca de una serie de ítems y hemos recabado información acerca de si los conocían previamente o no. A continuación explicamos el diseño y desarrollo del experimento, seguido de un análisis de los resultados obtenidos al emplear el conjunto de preferencias como entrada de los recomendadores.

6.3.1 Diseño y desarrollo

Para recabar un conjunto de votos en ausencia de sesgos de descubrimiento decidimos anular la capacidad de los usuarios de decidir sobre lo que votan y lo que no. Para ello, solicitamos a los usuarios de Crowdflower – una plataforma de crowdsourcing – que votaran una serie de ítems preseleccionados anteriormente, así como que indicaran si los conocían con anterioridad o no.

A continuación argumentamos cómo seleccionamos los ítems acerca de los cuales preguntamos a los usuarios, detallamos el diseño exacto de la tarea que pedimos que realizaran y explicamos brevemente algunas cuestiones implementativas.

6.3.1.1 Selección de los ítems

Para realizar la votación en ausencia de sesgos de descubrimiento seleccionamos aleatoriamente los ítems que debían votar los usuarios y se los mostramos para que los evaluaran. Esto implica que los usuarios no tenían por qué conocer previamente los ítems y necesitaban un cierto tiempo de evaluación. Con vistas a obtener el mayor número de votos por usuario es deseable que esta evaluación sea lo más rápida posible, prácticamente inmediata, lo cual limita el domino de los ítems, pues productos como libros o películas quedan descartados. Por ello, en el experimento empleamos el dominio de las canciones, ya que un usuario puede juzgar si le gusta o no una canción escuchando únicamente unos segundos de la misma.

Por otro lado, para seleccionar los ítems concretos es deseable una base de datos de pistas musicales lo más amplia posible en la que se puedan muestrear canciones aleatoriamente y obtener de ellas un audio para reproducir. La base de datos que empleamos para ello es la base de datos de Deezer[16], una aplicación que permite a los usuarios reproducir las canciones que desean, similar a Spotify o Last.fm.

La base de datos de Deezer contiene más de 40 millones de audios, pero algunos de ellos son diferentes versiones de la misma canción. En la práctica, esto implica que hay canciones con más probabilidad de ser muestreadas – las que más versiones presentan – y generalmente son las más conocidas. Esto puede producir un posible sesgo de descubrimiento, pero es sensato asumir que dicho sesgo es mucho menor que el que depende de las interacciones de los usuarios y, comparativamente, se puede considerar inapreciable.

Junto con su base de datos Deezer suministra una API para desarrolladores que permite realizar consultas a través de mensajes HTTP GET cuya respuesta se devuelve en formato JSON. En particular, es posible obtener a partir de un identificador numérico diversos datos de la canción, entre los que se encuentra la url de un extracto de 30 segundos. El hecho de que el identificador sea numérico resultó clave para seleccionar esta base de datos, pues otras más conocidas como MusicBrainz[17] reciben un número cifrado que no puede ser generado aleatoriamente. De igual forma, descartamos otras opciones, como Last.fm, porque no ofrecían la opción de conseguir un audio de la canción.

Para realizar el muestreo aleatorio es necesario calcular el número máximo que puede tomar el identificador de Deezer. No es una información que se facilite en la aplicación, por lo que la calculamos mediante un programa que realiza peticiones con identificadores cada vez más altos. Como los identificadores válidos tampoco son necesariamente consecutivos asumimos que el último identificador válido es aquel tras el cual los siguientes 10^{14} números no lo son. Una vez conocido este número seleccionamos los ítems generando aleatoriamente identificadores en el rango calculado – descartando aquellos inválidos – y descargando los datos de las canciones asociadas.

Mediante el procedimiento anterior muestreamos un total de 1100 canciones. El objetivo era conseguir más de 1000 canciones válidas, por lo que tomamos ese margen de cien en previsión

[16] https://www.deezer.com/
[17] https://musicbrainz.org/

de que algunos de los audios descargados pudieran estar dañados, ser ruidos, diálogos o canciones ofensivas. La forma de detectar dichos audios erróneos la explicamos a continuación en el diseño de la tarea.

6.3.1.2 Diseño de la tarea

Pese a que sería deseable obtener un conjunto de preferencias completo en el que se conociera toda la relevancia real, es decir, en el que estuviera presente el voto de todos los usuarios sobre todos los ítems, no es realista pedir a cada usuario que vote mil canciones de las cuales una gran mayoría seguramente no conozca y necesite tiempo para valorarlas.

Por ello, decidimos limitar el número de canciones votadas por cada usuario a 100, lo que implica que en torno a un 90% la relevancia real sigue estando oculta. Sin embargo, las canciones que vota cada usuario son seleccionadas aleatoriamente, lo que permite asumir que la relevancia observada es una muestra consistente y representativa de la relevancia real, pues los sesgos de descubrimiento son prácticamente inexistentes.

Las cien canciones asignadas a cada usuario se dividieron en grupos de diez de forma que cada grupo conformaba lo que en Crowdflower se denomina una "tarea". Una tarea representa la unidad mínima de lo que un usuario puede realizar y ser compensado por ello, en nuestro caso cada tarea era premiada con 5 céntimos. Esta división en tareas implica que los usuarios no podían únicamente valorar una canción y dejar el experimento – no si pretendían ser recompensados por ello – pero sí podían únicamente valorar diez. Pese a que pedimos a los usuarios que realizaran las diez tareas correspondientes a las cien canciones, algunos decidieron realizar menos por lo que eliminamos sus preferencias del conjunto final de votos y esperamos a que hubiera un número mínimo de usuarios que completaran todas las tareas. En nuestro caso interesaba tener más de 1000 usuarios.

Respecto a la forma de valorar cada canción, ofrecimos cinco opciones a los usuarios, de las cuales únicamente podían elegir una: cuatro hacían referencia al grado de relevancia de la canción para el usuario y la última servía para detectar aquellos audios que pudieran estar dañados o no ser canciones. De las cuatro posibles respuestas para evaluar el grado de relevancia, dos de ellas tenían un claro matiz positivo, aunque una de ellas más que la otra, ("me gusta realmente" y "es agradable y he disfrutado escuchándola"), otra era neutral ("me es indiferente") y la última era claramente negativa ("no me gusta"). Debido a la multitud de nacionalidades de los usuarios de Crowdflower, la parte escrita del experimento – las instrucciones y las preguntas – está redactada en inglés.

Junto con la pregunta acerca del voto del usuario por la canción incluimos otra pregunta en la que consultabamos al usuario acerca de si conocía la canción con anterioridad. El objetivo de esta pregunta es conseguir una muestra de la distribución de descubrimiento. Al igual que ocurre con la relevancia, el hecho de que el conjunto no es completo implica que se desconoce el valor de dicha distribución en una gran cantidad de ítems, pero la ausencia de sesgos permite aproximarla de forma fiable mediante la muestra.

En la Figura 14 se muestra la estructura presentada a los usuarios para valorar cada canción. Tal y como se observa, la estructura cuenta con un reproductor en la parte superior para poder escucharla. Para evitar ningún tipo de influencia no suministramos a los usuarios información alguna acerca del título o autor de la canción, sino que únicamente disponían del audio para juzgar su relevancia.

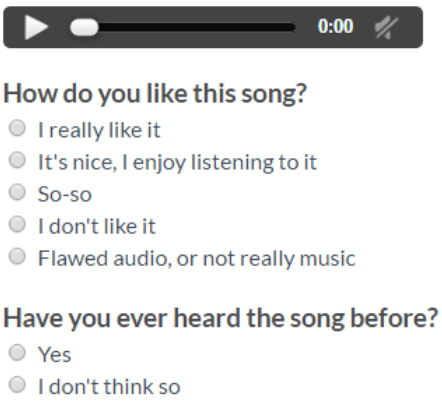

Figura 14. Estructura de las preguntas realizadas a los usuarios sobre cada canción.

En cada tarea, Crowdflower incluye una pestaña desplegable con las instrucciones del experimento. En nuestro caso dichas instrucciones, mostradas en la Figura 15, explicaban brevemente el experimento y lo que se requería de los usuarios.

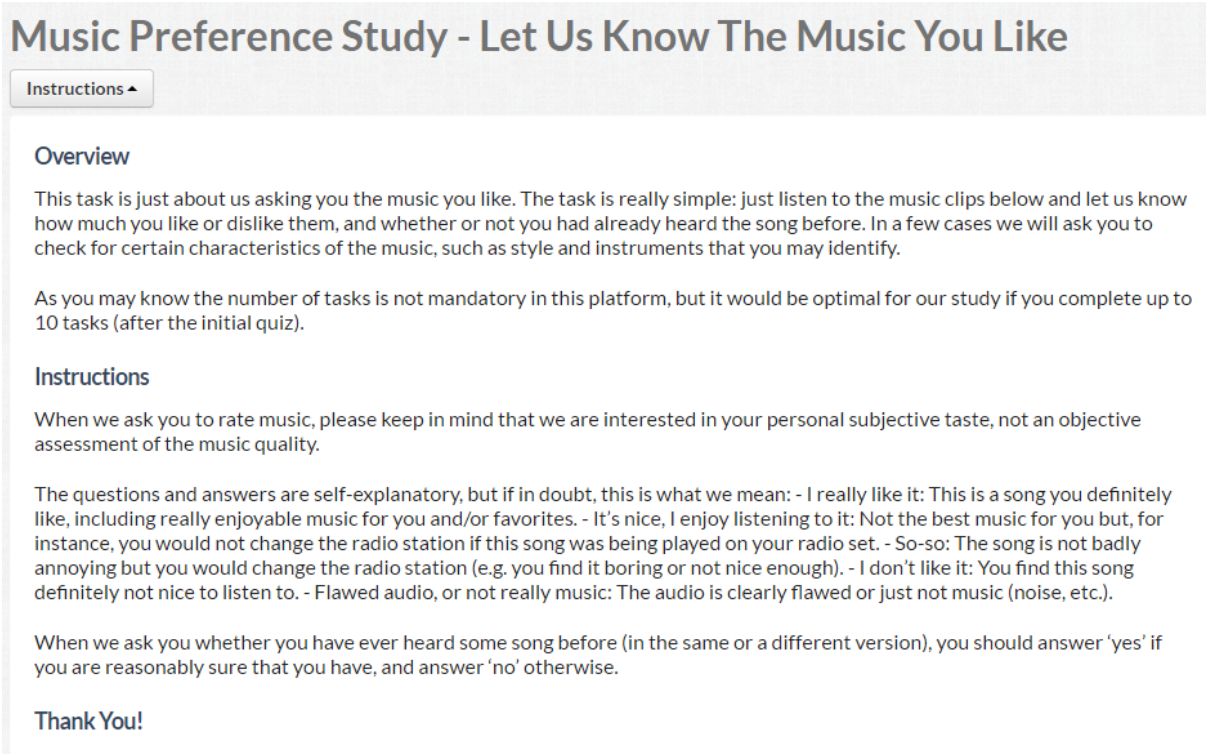

Figura 15. Instrucciones aportadas a los usuarios para realizar el experimento.

Una última cuestión al respecto del diseño de la tarea hace referencia a la forma de evitar usuarios que presentan un comportamiento aleatorio. Como en todo experimento con usuarios reales, existe la posibilidad de que algunos decidan no contestar con sinceridad y se limiten a contestar las preguntas al azar, sin escuchar la canción, para así minimizar el tiempo empleado y maximizar la ganancia. Las plataformas de crowdsourcing tienden a atraer a este tipo de usuarios, por lo que suelen suministrar una serie de elementos para detectar y prevenir el comportamiento aleatorio. Las herramientas que empleamos a este respecto son las siguientes:

- Mínimo de tiempo para la tarea. Establecimos un mínimo de 60 segundos para valorar las canciones de cada tarea. De esta forma, aquellos usuarios que seleccionen sus votos

rápidamente al azar y sin escuchar las canciones serán eliminados al intentar pasar a la siguiente tarea.

- Preguntas de control. Las preguntas de control suelen ser cuestiones con la misma estructura que las preguntas del experimento pero de las cuales se sabe la respuesta. Estas preguntas se intercalan con las preguntas reales de forma que si el usuario falla en contestarlas adecuadamente es penalizado o expulsado.

  En nuestro caso, el diseño de estas preguntas era un poco más complicado, pues cuando se pregunta acerca de los gustos de los usuarios no hay una única respuesta correcta. Por ello, establecimos como preguntas de control una serie de audios que estaban abiertamente dañados o no eran música, de forma que la respuesta correcta era indicar que el audio era erróneo.

  De igual forma, empleamos preguntas con otro formato en las que preguntábamos acerca de los elementos que aparecían en la canción o sobre el estilo de la música. Dos ejemplos de la estructura de estas preguntas se pueden observar en la Figura 16. En realidad, este tipo de preguntas más que preguntas de control hacían la labor de señuelos, pues pretendíamos que los usuarios de comportamiento aleatorio al detectar un formato distinto pensaran que esa era la pregunta de control, no prestaran atención al resto de canciones y fueran captados por la pregunta de control camuflada.

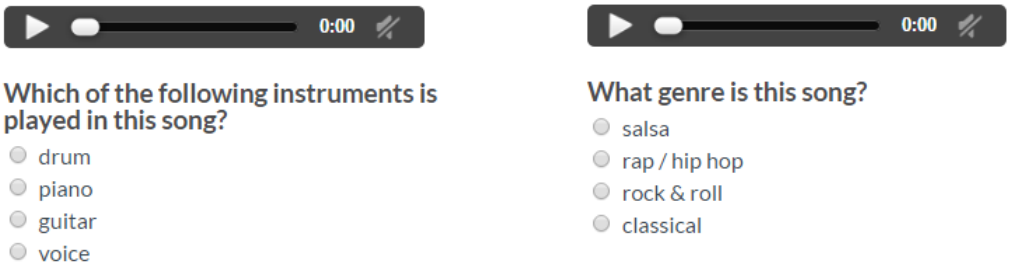

Figura 16. Dos ejemplos de preguntas señuelo.

  De esta forma, y tal y como se muestra en la Figura 17, en cada tarea se mostraban doce canciones, las diez de las que pedíamos realmente valoración, una pregunta de control con un audio falso y un señuelo de la forma de las preguntas de la Figura 16.

- Pese a las anteriores medidas, en los experimentos de prueba detectamos varios comportamientos anómalos entre los usuarios, como el hecho de que conocían un número extremadamente alto y poco verosímil de canciones. Esto sugería una forma de responder aleatoria, por lo que establecimos un límite superior del 50% de respuestas afirmativas válidas a la pregunta acerca de si se conocía previamente la canción. De esta forma, si un usuario afirmaba conocer más de la mitad de las canciones valoradas era expulsado.

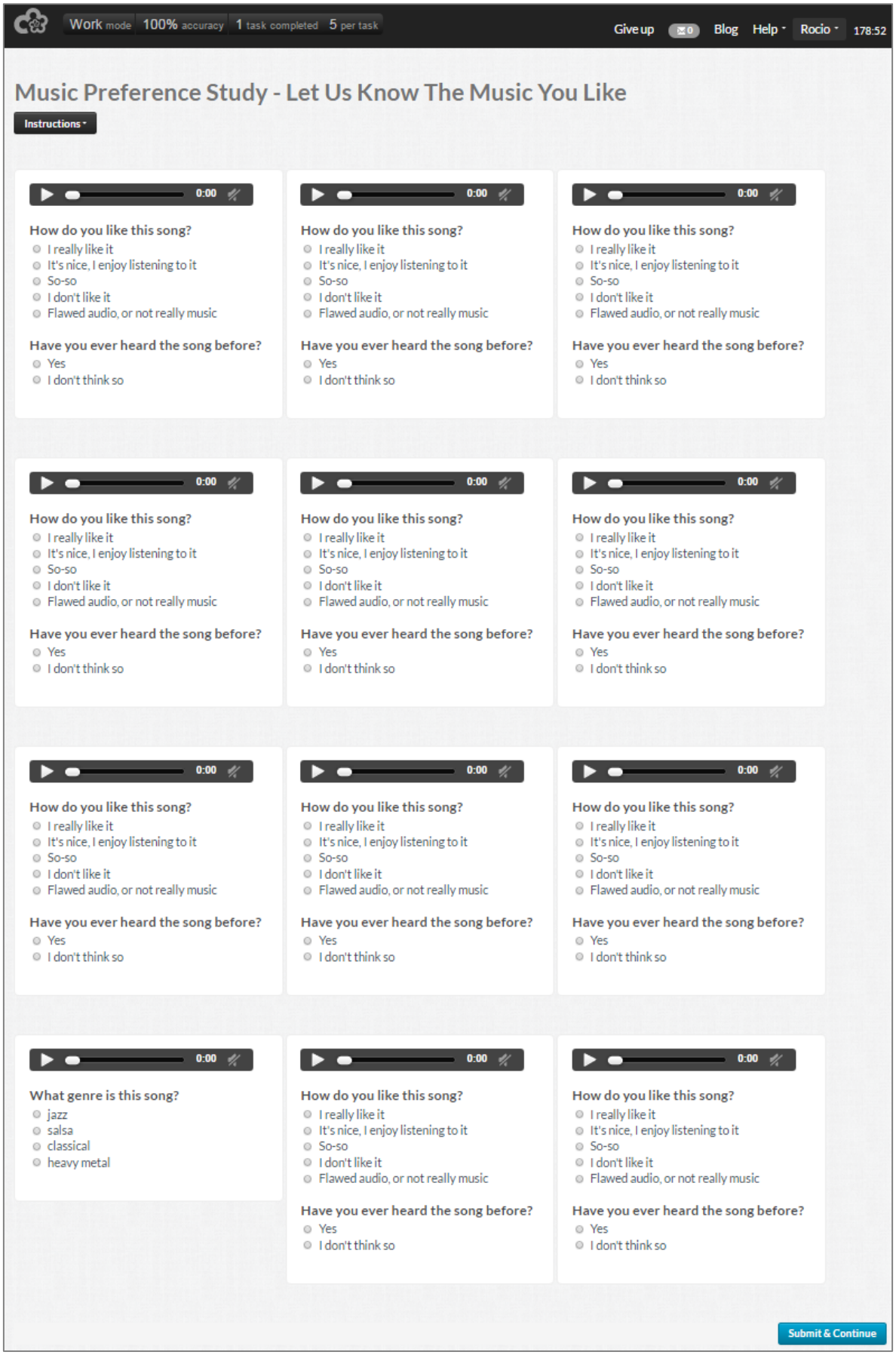

Figura 17. Estructura de la tarea requerida al usuario.

6.3.1.3 Implementación

Como ya hemos introducido anteriormente, el experimento ha sido desarrollado en Crowdflower, una plataforma de crowdsourcing enfocada en recabar "juicios" – terminología tomada de la propia plataforma – que, a priori, únicamente los seres humanos son capaces de realizar de forma óptima, como por ejemplo la clasificación, la extracción de información útil o el análisis, de imágenes, textos, tweets, páginas web, etc. De hecho, muchas veces el objetivo de dichos juicios es servir como conjunto de entrenamiento de algoritmos de aprendizaje automático que pretenden reproducir esas habilidades humanas. En nuestro caso, dichos juicios consisten en las opiniones de los usuarios acerca de un conjunto de canciones.

En Crowdflower existen dos roles o tipos de usuarios, los *customers* o usuarios que solicitan a la plataforma la realización de un trabajo que implica recabar un conjunto de juicios – nuestro papel en este experimento – y los *contributors* o usuarios que realizan los juicios a cambio de una cierta retribución aportada por el *customer*. En nuestro caso, los *contributors* son los usuarios cuyos votos queremos recabar.

En la Figura 18 se muestra un esquema con los pasos que se llevan a cabo en Crowdflower para realizar un trabajo. En la etapa inicial (paso 0) los *customers* configurar el experimento para indicar las unidades mínimas concretas de las que se requiere recabar juicios – en nuestro caso las canciones –, dichas unidades son almacenadas en la base de datos por el motor de Crowdflower. Una vez realizada esta configuración, se agrupan las unidades para formar tareas (pasos 1 y 2) que se envían a los *contributors* (paso 3) para que las realicen. Cada tarea completada válida se retorna al sistema (paso 4) que almacena los juicios realizados en ella en la base de datos (paso 5). Si se detecta alguna anomalía que hace sospechar acerca de la validez de los juicios del usuario, los juicios se siguen almacenando – indicando que son cuestionables – pero no se retribuye a dicho usuario. Una vez recabados todos los juicios deseados acerca de cada unidad el trabajo finaliza y los *customers* pueden acceder a la base de datos para consultar y descargar los resultados (paso 6).

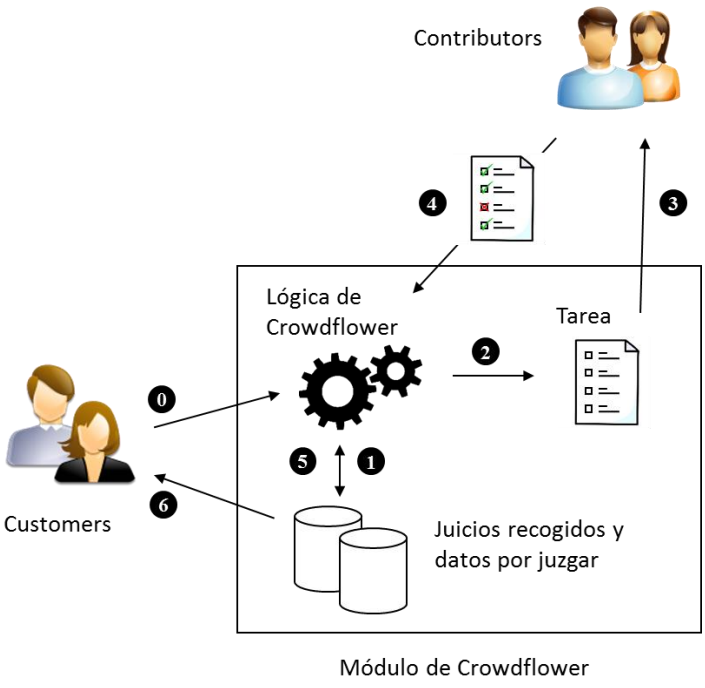

Figura 18. Esquema del funcionamiento de Crowdflower.

Como Crowdflower se especializa en conseguir juicios sobre tareas que suelen referirse a la clasificación o limpieza de datos, el objetivo es conseguir la mayor cantidad de juicios en un cierto tiempo, y no se presta especial importancia a quién hace qué. Así, la agrupación en tareas es fija para todos los usuarios, de forma que si dos unidades son incluidas en la misma tarea se mostrarán juntas a todos los usuarios hasta alcanzar el número de juicios deseado. Una vez alcanzada dicha cifra se formara otra tarea con las unidades que queden sin juzgar y será esa nueva tarea la que se entregue a los nuevos usuarios.

En nuestro experimento, esto supone un coeficiente de clustering de la matriz de ratings muy alto, pues los usuarios que realizan el experimento al mismo tiempo votan las mismas canciones. Así, si dos usuarios han votado la misma canción la probabilidad de que coincidan en otras era muy alta, lo cual produce un sesgo a favor de los algoritmos de recomendación basados en filtrado colaborativo que hace que obtengan una precisión anormalmente alta.

Para sustituir la agrupación fija de Crowdflower por una aleatoria que evite sesgos de este tipo modificamos el esquema de Crowdflower para que siguiera el proceso descrito en la Figura 19. Desarrollamos un servlet externo al que se llama desde el lado de los *contributors* y que modifica la tarea incluyendo otras canciones seleccionadas de forma aleatoria. Para esta implementación emplemaos un editor de código de Javascript proporcionado por Crowdflower que permite que dicho código se ejecute al cargar la página en el lado del *contributor*. Mediante este código Javascript modificamos el código html de la página para incluir los scripts que contactaban con el servidor web externo. De esta forma, tras el envío de la tarea al usuario (paso 3) se ejecuta el código Javascript que envía la tarea al servlet (paso 4). Dicho servlet consulta la base de datos externa para constatar qué canciones ya han sido votadas por cada usuario y elige aleatoriamente diez con las que modifica la tarea (paso 5) para posteriormente devolverla al usuario (paso 6). A partir de este punto el funcionamiento es el natural de Crowdflower: el usuario devuelve los juicios y estos son almacenados por la plataforma.

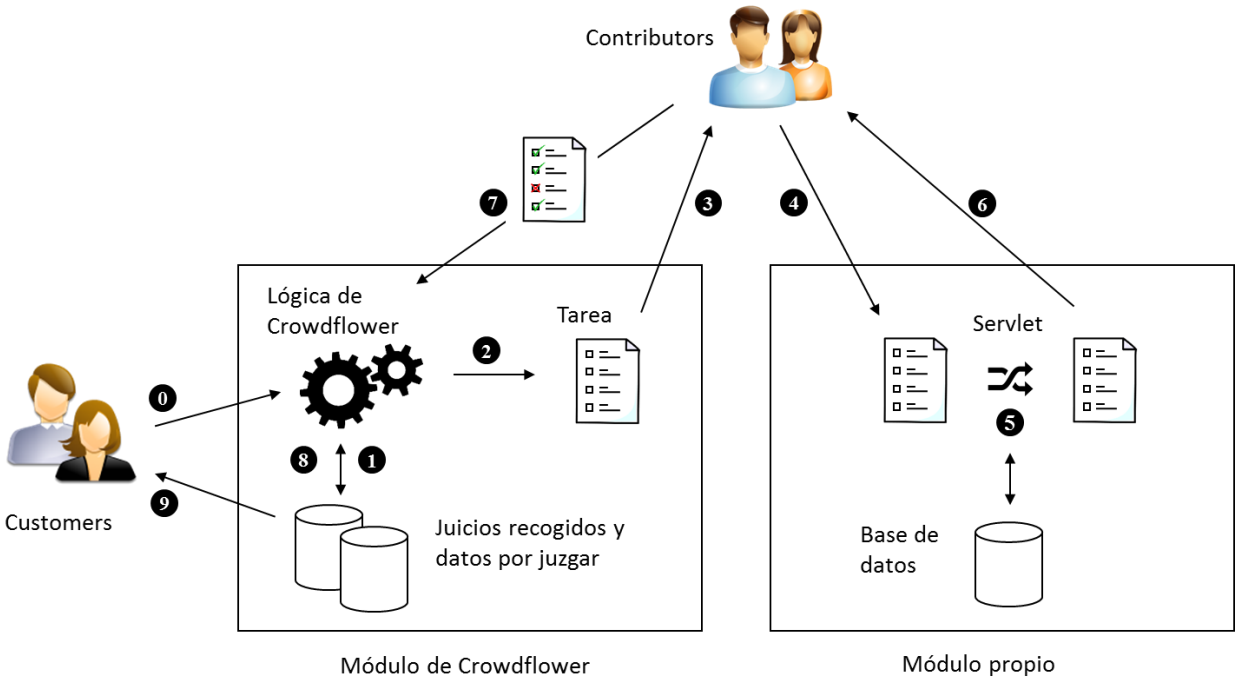

Figura 19. Esquema del sistema implementado para el desarrollo del experimento.

Cabe destacar que en el paso 7, cuando el usuario devuelve la tarea a Crowdflower, también se realiza una notificación al servlet externo notificando que dichos juicios han sido completados. Así, si el mismo usuario solicita otra tarea se escogerán otras canciones y no las mismas.

Así mismo, en la modificación del html de la tarea incluimos campos ocultos para identificar las canciones que están siendo juzgadas, es decir, para saber a qué canción hacen referencia los juicios almacenados en Crowdflower. También incluimos otros campos ocultos para almacenar el tiempo que se dedica a cada canción y el orden en el que se votan.

Mediante el enlace a pie de página[18] se puede acceder, en el rol de *contributor*, al cuestionario desarrollado. Es un cuestionario configurado para que se encuentre en modo de prueba, de forma que únicamente se puede acceder desde la dirección anterior y las tareas realizadas no se financian. Para realizarlo es necesario registrarse en Crowdflower.

Si se accede al cuestionario se observará que la primera página muestra una serie de preguntas de control en el formato mostrado en la Figura 16. Esta página, denominada *Quiz mode*, es una herramienta de Crowdflower que no es posible eliminar y cuya finalidad es comprobar la habilidad de los usuarios para realizar las tareas del trabajo. En nuestro caso, las tareas no requieren habilidad alguna pero su realización se impone desde Crowdflower, que conforma dicha tarea a partir de las preguntas de control configuradas internamente. Una vez contestado correctamente – únicamente se permiten un error – se pasa al modo *Work mode*, en el que se inicia el cuestionario en sí con las diez tareas explicadas anteriormente.

6.3.2 Resultados

Tras la implementación y configuración de la plataforma, en junio de 2015 indicamos a Crowdflower que iniciara la recopilación de juicios, la cual finalizó tras 8 meses, en febrero de 2016. El experimento no estuvo funcionando todo este tiempo, pues fue necesario interrumpir dicha recopilación de datos por más de un mes debido a que el servidor en el que se encontraba el servlet que permitía aleatorizar las canciones fue dañado y hubo que migrar la funcionalidad a otra máquina.

A continuación exponemos, en primer lugar, las características del conjunto de datos recabado y, posteriormente, mostramos y explicamos los resultados obtenidos al ejecutar los recomendadores sobre dicho conjunto.

6.3.2.1 Análisis del conjunto de datos

Una vez recabados los datos realizamos una limpieza de los mismos para conformar el conjunto definitivo. Así, por un lado seleccionamos únicamente aquellos usuarios que habían realizado diez tareas y, por otro, eliminamos los audios marcados como dañados por al menos diez usuarios. Mediante este último procedimiento fueron eliminados un total de 17 ítems. También eliminamos otras 5 canciones que no habían llegado a ser votadas por más de cincuenta usuarios, para evitar sesgos.

Las dimensiones del conjunto de datos definitivo pueden observarse en la Tabla 4.

[18] https://tasks.crowdflower.com/channels/cf_internal/jobs/914209/work?secret=ifOJA7lFaDon2%2BEtNtwVUi%2Fk20JZagwLl8XPFrpFU6bG

Estadísticas	
Nº. de usuarios	1.072
Nº. de ítems	1.084
Nº. de ratings	105.348
Densidad	9,1%
Densidad relevante	2,6%

Tabla 4. Dimensiones del conjunto de preferencias obtenido a partir de las votaciones de los usuarios de Crowdflower.

En la Figura 20 se muestra el porcentaje de usuarios que ha seleccionado cada opción de las dos preguntas realizadas. Respecto a la opinión sobre la canción, observamos que, como cabría esperar de una muestra aleatoria de canciones, la mayor parte de los juicios implican que la canción no le ha gustado al usuario o que le es indiferente (en torno a un 70%), mientras que muy pocas veces ha gustado realmente. De nuevo concordando con esta selección aleatoria de canciones, la mayoría de respuestas a la pregunta sobre si se conocía o no la canción con anterioridad son negativas y únicamente en un 11,37% de los casos los usuarios conocían el audio.

Para los posteriores análisis realizamos una serie de simplificaciones sobre los datos. En particular, para ajustar el experimento a la relevancia binaria del análisis teórico consideramos que una canción es relevante para el usuario si ha votado cualquiera de las dos opciones que implican una opinión positiva (*It's nice*, *I enjoy listening to it* o *I really like it*). El resto de opciones, incluida la de considerar que el audio no es válido, se considera un voto no relevante.

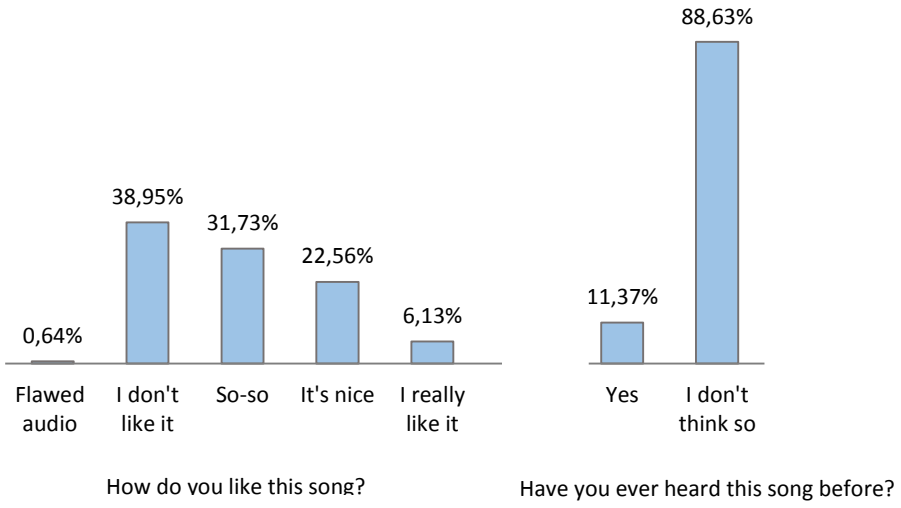

Figura 20. Porcentaje de usuarios que han votado cada una de las opciones a las dos preguntas realizadas.

Tal y como pretendiamos con el diseño del experimento, al incluir en los datos recabados información acerca de la distribución de descubrimiento es posible simular una situación típica de evaluación offline. Así, consideramos que los votos de los usuarios que ya conocían el ítem se corresponden con la relevancia observada, mientras que el resto de votos conforman una muestra representativa de la relevancia real no observada. En esta situación, la popularidad de un ítem es el número de usuarios que ya lo conocían, no el número de votos totales, porque estos son aleatorios.

De acuerdo con esta interpretación, las distribuciones de popularidad (número de usuarios que conoce cada ítem) y relevancia (número de usuarios que considera relevante cada ítem, lo conozcan o no) del conjunto de datos se muestran en la Figura 21, en donde los ítems del eje x se encuentra ordenados por su popularidad. Vemos que la curva de descubrimiento presenta una forma bastante menos segada que las que se suelen encontrar en otros conjuntos de datos, como los que estudiamos en el capítulo 2. Así, es cierto que hay unos pocos ítems muy conocidos, pero una vez superados dichos ítems la curva desciende casi linealmente.

Respecto a la relación entre las distribuciones de descubrimiento y relevancia, en la Figura 21 se observa una cierta dependencia positiva que viene confirmada por la correlación de Pearson entre ambas distribuciones, cuyo valor es 0.58.

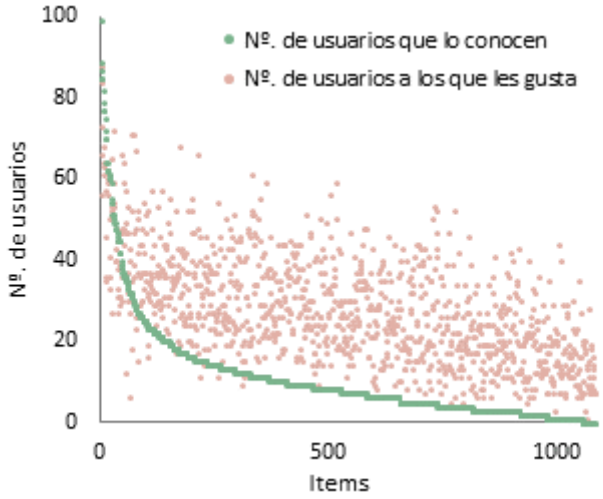

Figura 21. Distribución de popularidad (usuarios que conocen cada ítem) del conjunto de Crowdflower junto con la distribución de relevancia real (usuarios a los que les gusta). El eje *x* se corresponde con los ítems ordenados por su popularidad.

6.3.2.2 Análisis de las recomendaciones

Tal y como comentamos anteriormente, la distinción entre canciones conocidas y no conocidas permite recrear una situación de evaluación offline en la que, además de la relevancia observada, se conoce una muestra no sesgada de la relevancia real. Esto permite calcular tanto la precisión observada como la real.

Así, para calcular ambas precisiones dividimos los datos observados – aquellos votos en los que los usuarios han indicado que conocían la canción – en un conjunto de entrenamiento y otro de test mediante una partición aleatoria y empleamos el conjunto de entrenamiento como entrada de los recomendadores (aleatorio, popularidad total y popularidad relevante). En esta situación, la precisión observada se corresponde con la obtenida al evaluar con el conjunto de test, mientras que la precisión real se calcula evaluando tanto con los datos del conjunto de test como con los no observados.

En la Figura 22.a se muestran las precisiones – en la primera posición – obtenidas por los recomendadores al seguir la metodología anterior y empleando como tasa de entrenamiento el valor 0.5. Para evitar los efectos de la varianza de la partición, dichas precisiones se han promediado sobre cien particiones. De igual forma, el recomendador aleatorio ha sido ejecutado diez veces sobre cada partición.

Se observa que tanto en precisión real cómo en observada la comparación entre los recomendadores se mantiene. Además, el orden es el que cabría esperar: la precisión más alta la obtiene popularidad relevante, seguida por popularidad total y por último, y con una gran diferencia, se encuentra el recomendador aleatorio. Llama la atención que esta diferencia es mayor en el caso de la precisión observada, lo cual indica que llega un momento en que los recomendadores por popularidad no son capaces de sacar provecho del aumento en el número de datos con los que se evalúa, o no de forma proporcional al aumento de la probabilidad de acertar al elegir un ítem al azar.

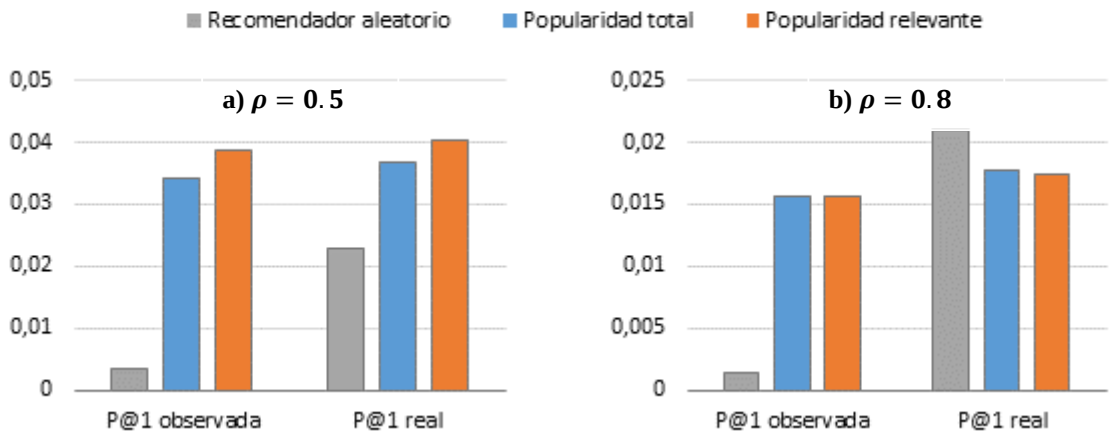

Figura 22. Precisión observada y real en la primera posición de la recomendación producida por los recomendadores aleatorio, popularidad total y popularidad relevante, al ejecutarlos sobre el conjunto de preferencias observadas, es decir, aquellas en las que el usuario ha indicado que conocía previamente la canción. La gráfica de la izquierda (a) se corresponde con una partición aleatoria cuya tasa de entrenamiento ρ es 0.5 y la de la derecha (b) con una tasa de entrenamiento 0.8.

Empleando una tasa de entrenamiento de 0.5, los resultados se encuadran dentro de la normalidad y aportan seguridad al observar que ambas precisiones concuerdan. Sin embargo, la situación cambia notable y sorprendentemente al emplear como tasa de entrenamiento 0.8, situación que se muestra en la Figura 22.b. En este caso, la precisión observada presenta un comportamiento semejante al caso anterior, destacando únicamente el hecho de que ambas precisiones se igualan. Es la precisión real la que llama la atención, pues en ella el recomendador aleatorio supera a ambas popularidades. Es decir, en una situación típica de evaluación offline con estos datos observados se estaría obteniendo que los recomendadores por popularidad son muy superiores al recomendador aleatorio – cómo de hecho se suele obtener en todos los conjuntos de datos – mientras que la realidad es justamente la contraria: emplear el recomendador aleatorio supone acertar más con los gustos del usuario.

Esta situación se encuadra dentro del punto 6.2.3 del análisis teórico, es decir, la probabilidad de votar es independiente del resto de variables – en particular de la relevancia – y el descubrimiento no es independiente ni de la relevancia ni del ítem, como se puede observar en la Figura 21. Ya habíamos anticipado que en este tipo de situaciones cualquier comportamiento es posible y que, además, dicho comportamiento podía variar también con el valor de ρ.

Como mencionamos anteriormente, existe una relativamente alta correlación entre descubrimiento y relevancia de 0.58. Esto implica que las popularidades, al ordenar por $p(seen|\, i)$ y $p(seen, rel|\, i)$, generan un ránking muy similar al obtenido si se ordena por relevancia, en especial en las primeras posiciones que son las que más influyen en la precisión.

Por tanto, el fenómeno que se está produciendo no es que las popularidades no ordenen por la relevancia real de los ítems, sino que ordenar por dicha relevancia produce un ránking peor – respecto a la precisión real – que el generado de forma aleatoria. Para comprobar este comportamiento nos remitimos al cociente por el que realmente se deberían ordenar los ítems para obtener la máxima precisión real:

$$C(i) = \frac{p(rel|i)\big(1 - \rho\ C\ p(seen|\ rel, i)\big)}{1 - \rho\ C\ p(seen|\ i)}$$

En la Figura 23 se muestra el valor de dicho cociente – empleando las tasas de entrenamiento 0.5 (fila superior) y 0.8 (fila inferior) – para los diez ítems con más votos (columna izquierda) y con más votos relevantes (columna derecha) que serán los que en la práctica recomienden las popularidades total y relevante, respectivamente. Se muestra también una línea con el valor del promedio de $C(i)$ sobre todos los ítems, que representa la precisión que de media alcanzará el recomendador aleatorio.

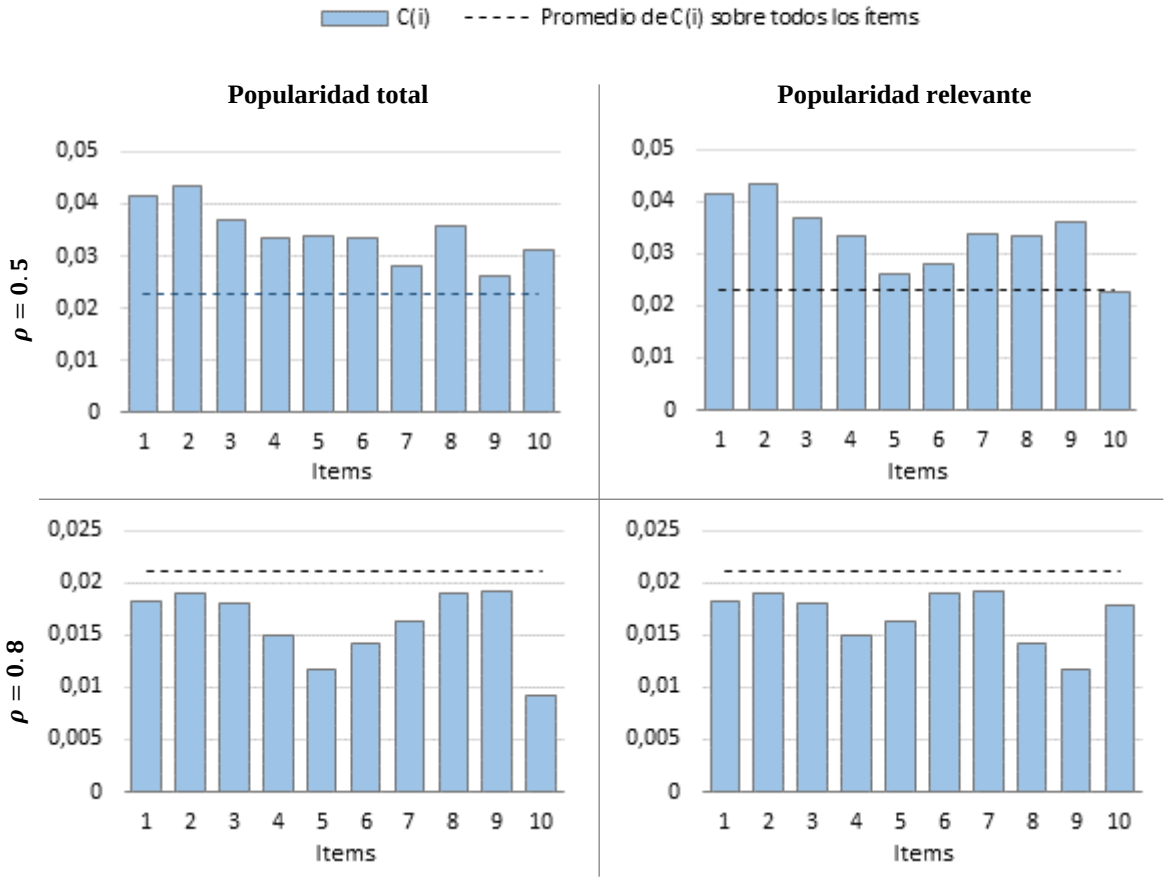

Figura 23. Valor del cociente $C(i)$ – empleando las tasas de entrenamiento 0.5 (fila superior) y 0.8 (fila inferior) – para los diez ítems con más votos (columna izquierda) y con más votos relevantes (columna derecha) que serán los que recomienden las popularidades total y relevante, respectivamente. Se muestra también una línea con el valor del promedio de $C(i)$ sobre todos los ítems, que representa la precisión que de media alcanzará el recomendador aleatorio.

Se observa claramente que para una tasa de entrenamiento 0.5 los primeros diez ítems presentan un cociente $C(i)$ superior al promedio, mientras que para una tasa de entrenamiento de 0.8 todos se encuentran por debajo. Es decir, en el primer caso los ítems que las popularidades están recomendando aportan una precisión mayor que el promedio – están correctamente ordenados respecto a la mayor parte de ítems – mientras que al aumentar la tasa

de entrenamiento dicha aportación disminuye frente al resto de ítems que pasan a ser mejores candidatos.

Intuitivamente, lo que ocurre es que los ítems más conocidos saturan su relevancia – llegan a todos los usuarios a los que potencialmente les gustan – por lo que aquellos que no los conocen es porque generalmente no les son relevantes. De esta forma, al ampliar la tasa de entrenamiento y eliminar del conjunto de test – para pasarlos a entrenamiento – votos de usuarios que conocen el ítem, quedan únicamente en dicho conjunto los votos de los usuarios que no lo conocían que suelen considerar no relevantes a estos ítems tan populares.

Otro aspecto que cabe destacar observando la Figura 23 es que los cuatro primeros ítems son los mismos en ambas popularidades, motivo por el cual presentan una precisión tan similar. Este orden tan similar se deduce de la alta correlación entre las distribuciones de descubrimiento y relevancia, en especial en los primeros ítems.

También resulta interesante constatar cómo resulta la evaluación cuando se emplea como relevancia observada todo el conjunto de preferencias, con independencia de si la canción era conocida o no. En este caso, toda la relevancia real de la que disponíamos en la interpretación anterior es observada y, por tanto, únicamente tiene sentido considerar la precisión observada. Para la precisión real habría que conocer la opinión de los usuarios sobre aquellos ítems que no han votado.

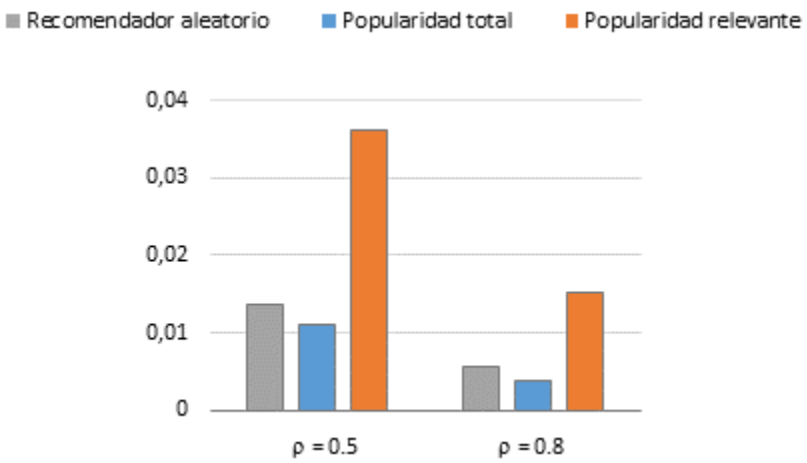

Figura 24. Precisión (observada) obtenida al tomar como entrada de los recomendadores todo el conjunto de preferencias, con independencia de si la canción era conocida o no por el usuario.

Así, en la Figura 24 se muestra la precisión (observada) de los tres recomendadores para las tasas de entrenamiento 0.5 y 0.8. Como cabría esperar, popularidad relevante es muy superior en ambos casos. Sin embargo, llama la atención que popularidad total se encuentra por debajo del recomendador aleatorio con ambas tasas de entrenamiento, cuando cabría esperar que se encontrara al mismo nivel, dado que en este caso el número de votos que recibe un ítem es aleatorio.

Lo que ocurre es un fenómeno que no hemos considerado hasta el momento, que la partición no conserve, en los conjuntos de entrenamiento y test, la distribución de los datos originales. Esta hipótesis la introdujimos al inicio de la sección 5.1, al considerar la partición aleatoria, y la hemos mantenido a lo largo de todo el desarrollo teórico.

De hecho, en todos los ejemplos empíricos anteriores dicha hipótesis se cumple, pues la curva de popularidad es lo suficientemente sesgada para que, al dividir los datos, un mayor número de votos en entrenamiento implique también un mayor número en test, al menos en los ítems más populares que son los que realmente influyen. Sin embargo, en este caso el número de votos por ítem depende de a cuantos usuarios les haya sido asignado para votarlo, proceso que es aleatorio. Por ello, la distribución de popularidad es muy uniforme y pueden producirse inversiones.

En la sección 7.3 estudiamos más en detalle cómo y porqué se produce este fenómeno, que está relacionado con la diferencia de varianzas entre las distribuciones de los datos y la partición. En este punto, únicamente cabe señalar que la inversión entre el recomendador aleatorio y popularidad total no es especialmente alarmante, pues ya esperábamos que – como mucho – se encontraran al mismo nivel, ya que en este caso lo popular que sea un ítem deja de aportar información. Sin embargo, este fenómeno nos lleva a considerar la posibilidad de que, en ciertas situaciones, el protocolo de partición pueda alterar completamente la precisión de los recomendadores. Estas consideraciones acerca de la influencia del protocolo de partición se estudian brevemente a continuación, en el siguiente capítulo.

7. Influencia del protocolo de partición

Hasta el momento hemos considerado una partición aleatoria que, además, mantiene la distribución de los datos originales en los conjuntos de training y test. Sin embargo, existen otras muchas formas de dividir los datos (partición temporal, partición usuario a usuario, etc.) y el resultado de la evaluación puede variar considerablemente de unas a otras.

En esta sección generalizamos la expresión analítica de las secciones anteriores para considerar un protocolo de partición menos específico. Sin embargo, la complejidad de dicha fórmula dificulta considerablemente un análisis formal de la influencia de cada tipo de partición, que queda fuera del alcance del presente trabajo. En su lugar, vamos a enfocarnos a constatar empíricamente esta influencia, aportando para ello diversos ejemplos en los que cambiar el protocolo de partición altera completamente el resultado de la evaluación y analizando los posibles factores que pueden estar interviniendo en cada caso.

7.1 Caracterización analítica

En el caso de la partición aleatoria todos los ratings tienen la misma probabilidad de ser asignados al conjunto de entrenamiento, $\rho = p(training|rate, i, u)$, con independencia del ítem o del usuario al que hacen referencia. En este estudio de la influencia de la partición, sin embargo, consideramos un protocolo de partición más genérico en el que el ítem sí puede influir. Respecto al usuario, mantenemos su independencia por coherencia con las hipótesis acerca de los recomendadores, en los que dicho usuario no influye.

De esta forma, en lugar de una única tasa de entrenamiento ρ, tenemos una tasa ρ_i para cada ítem:

$$\rho_i = p(training|rate, i, u)$$

Esta generalización del tipo de partición sigue produciendo una partición aleatoria – cada rating es asignado aleatoriamente a un conjunto u a otro – pero en este caso dicha partición sigue una distribución multinomial, por contraste con la partición aleatoria clásica que presenta una distribución binomial.

Con estas nuevas hipótesis, las ecuaciones 10 y 11 de las que se partía inicialmente en el estudio de la sección anterior resultarían de la siguiente forma:

$$\mathbb{E}[\bar{P}@1|R] = \sum_{k=1}^{n} \left(1 - \rho_{i_k}\right) p(rate|rel, i_k, R)\; p(rel|i_k, R) \prod_{j=1}^{k-1} \rho_{i_j}\; p\left(rate\middle|i_j, R\right)$$

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} \left(1 - \rho_{i_k}\, p(rate|rel, i_k, R)\right)\; p(rel|i_k, R) \prod_{j=1}^{k-1} \rho_{i_j}\; p\left(rate\middle|i_j, R\right)$$

Cabe destacar que para emplear dichas fórmulas se han de mantener las hipótesis que se plantearon a la hora de desarrollarlas: independencia del usuario, existencia de un único ránking e independencia de la variable $rate$ para ítems distintos.

En esta situación, estudiar cómo influye la distribución de las tasas de entrenamiento ρ_i en las fórmulas no es sencillo y se aleja del objetivo de este trabajo. Sin embargo, si resulta interesante observar empíricamente el efecto de otros tipos de particiones, como la partición temporal.

7.2 Partición temporal

La división de los datos en entrenamiento y test mediante una partición temporal únicamente es posible sobre votos de los que se conozca el momento en que fueron realizados. En esta situación, el protocolo consiste en asignar al conjunto de entrenamiento los votos que primero se han realizado dejando para el conjunto de test los más tardíos. Este planteamiento es más cercano a las situaciones reales de recomendación, en las que se emplean los datos de interacción pasados para tratar de predecir los gustos futuros.

También en la partición temporal existe el concepto de tasa de entrenamiento, al que nuevamente denominamos ρ, y que hace referencia al porcentaje de votos que son asignados a entrenamiento.

Aunque cabría considerar la precisión real, dado que los conjuntos con los que vamos a trabajar únicamente contienen información acerca de la relevancia observada nos vamos a limitar a considerar la precisión observada. El único conjunto que presenta información acerca de la relevancia real es el generado mediante el cuestionario a los usuarios de Crowdflower, pero en dicho conjunto el momento en que se realizó cada voto no es representativo, ya que fue seleccionado de forma aleatoria.

En esta situación, nos interesa contrastar si el empleo de una partición temporal puede afectar a la efectividad (observada) de la recomendación por popularidad. Intuitivamente, uno de los principales factores que puede influir en dicha efectividad es la velocidad a la que se propagan los ítems, es decir, la velocidad a la que obtienen votos. Así, generalmente la curva de evolución de un ítem sigue una distribución similar a una función logística como la que se observa en la Figura 25.

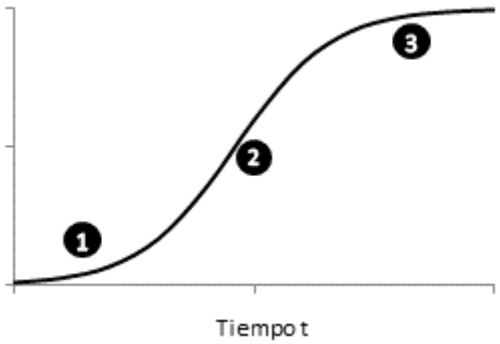

Figura 25. Forma de una curva logística.

Cuando es creado, pocos usuarios lo conocen y puntúan, por lo que su evolución inicial – en cuanto al número de votos – es lenta (punto 1). Según pasa el tiempo empieza a ser más conocido, por lo que la velocidad de propagación es mayor, llega a más usuarios y consigue

votos más rápido (punto 2). Por último, en la fase final el ítem se estabiliza (punto 3), bien porque todos los usuarios ya lo conocen o bien porque los que lo hacen ya no hablan de él. La forma de la curva de la Figura 25, en particular la rapidez con la que crece y se estabiliza, es clave a la hora de recomendar por popularidad, especialmente cuando la evaluación se realiza mediante una partición temporal. Así, si suponemos que un cierto ítem presenta una evolución en el número de votos similar a la indicada en la Figura 25, en función del punto por el que divida la partición temporal los votos de dicho ítem podemos tener situaciones muy distintas. Si lo hace en la fase inicial (punto 1), el ítem será considerado poco popular en entrenamiento, cuando realmente presenta gran cantidad de votos en test y podría ser óptimo recomendarlo. Mientras que si la división es en la fase final de estabilización (punto 3), será considerado muy popular pero al recomendarlo no quedarán prácticamente votos en test y su precisión será baja.

Resulta claro, por tanto, que la situación que mayor correlación presenta entre entrenamiento y test, y por tanto la que favorece a la recomendación por popularidad en términos de precisión observada, es aquella en la que la división temporal se produce en torno a la mitad de la fase de crecimiento rápido de la curva (punto 2). Sin embargo, en un entorno en el que los ítems se propagan muy rápido – como puede ser Twitter – esta fase de crecimiento puede durar muy poco tiempo y, con ello, la probabilidad de que la partición divida a todos los ítems por dicho punto ser muy baja. Especialmente si tenemos en cuenta que los puntos de inicio de los ítems varían, así como la altura que alcanzan.

Para constatar este fenómeno, en esta sección comparamos empíricamente los rendimientos de las dos popularidades – total y relevante – con el del recomendador aleatorio, cuando se emplea para dividir el conjunto de datos una partición temporal. Por un lado, empleamos datos provenientes de usuarios reales, más concretamente de los conjuntos de Netflix y Twitter. Por otro, para estudiar de forma más sistemática la influencia de la velocidad, utilizamos datos sintéticos simulados en los que los ítems crecen a una cierta velocidad.

En los conjuntos de datos anteriores, en los que existen muchos ítems con prácticamente ningún voto, emplear el recomendador aleatorio puro como medida de efectividad neutra no es una buena metodología. El motivo es que la precisión de una recomendación aleatoria tiende a cero si la cola de popularidad se extiende indefinidamente añadiendo ítems poco populares, mientras que la precisión de popularidad en dicho caso permanece constante. Por ello, como referencia más significativa vamos a considerar un hibrido entre popularidad y el recomendador aleatorio que consiste en seleccionar aleatoriamente el ítem a recomendar entre los top k más populares.

7.2.1 Datos reales

Para comprobar el efecto de la partición temporal al evaluar los recomendadores sobre datos provenientes de usuarios reales hemos seleccionado dos conjuntos, el conjunto de datos de Netflix y un subconjunto de interacciones de Twitter recabado por un compañero del grupo *Information Retrieval Group*.

7.2.1.1 Netflix

Recordamos que Netflix es un conjunto de votos de usuarios sobre películas cuyas dimensiones pueden consultarse en la Tabla 1 de la sección 2.2. A diferencia de MovieLens, Netflix contiene información acerca del momento en el que se realizaron los votos, por lo que se ajusta a los objetivos de este análisis.

En la Figura 26 se muestra el resultado de evaluar los recomendadores sobre los datos de Netflix empleando para ello una partición temporal de tasas de entrenamiento 0.5 (fila superior) y 0.8 (fila inferior). En las gráficas, el eje *x* representa el número de ítems entre los que pueden elegir los recomendadores a la hora de evaluar, ordenados según su popularidad. Así, los valores correspondientes al valor k del eje *x* son las precisiones observadas de los recomendadores cuando únicamente pueden recomendar los top k más populares. Lógicamente, para comparar con popularidad total los ítems se encuentran ordenados según el número de votos totales (columna izquierda) mientras que para comparar con popularidad relevantes se ordenan por el número de votos relevantes (columna derecha). Junto a las precisiones de los recomendadores se incluye una curva que representa su cobertura, esto es, la evolución del número de usuarios a los que se les ha podido recomendar. Dicha curva aparece en color verde y se corresponde con el eje y derecho, medido en cientos de miles de usuarios. Observamos que es creciente, es decir, que para valores de k pequeños existen usuarios que han votado todos los k ítems más populares, por lo que si dichos votos caen todos en entrenamiento no es posible realizar una recomendación a esos usuarios. En ese caso, descartamos dichos usuarios del cómputo de la precisión, como si no estuvieran presentes. Al aumentar el número de ítems se va haciendo menos frecuente la existencia de este tipo de usuarios que los han votado todos, por lo que la cobertura también se incrementa.

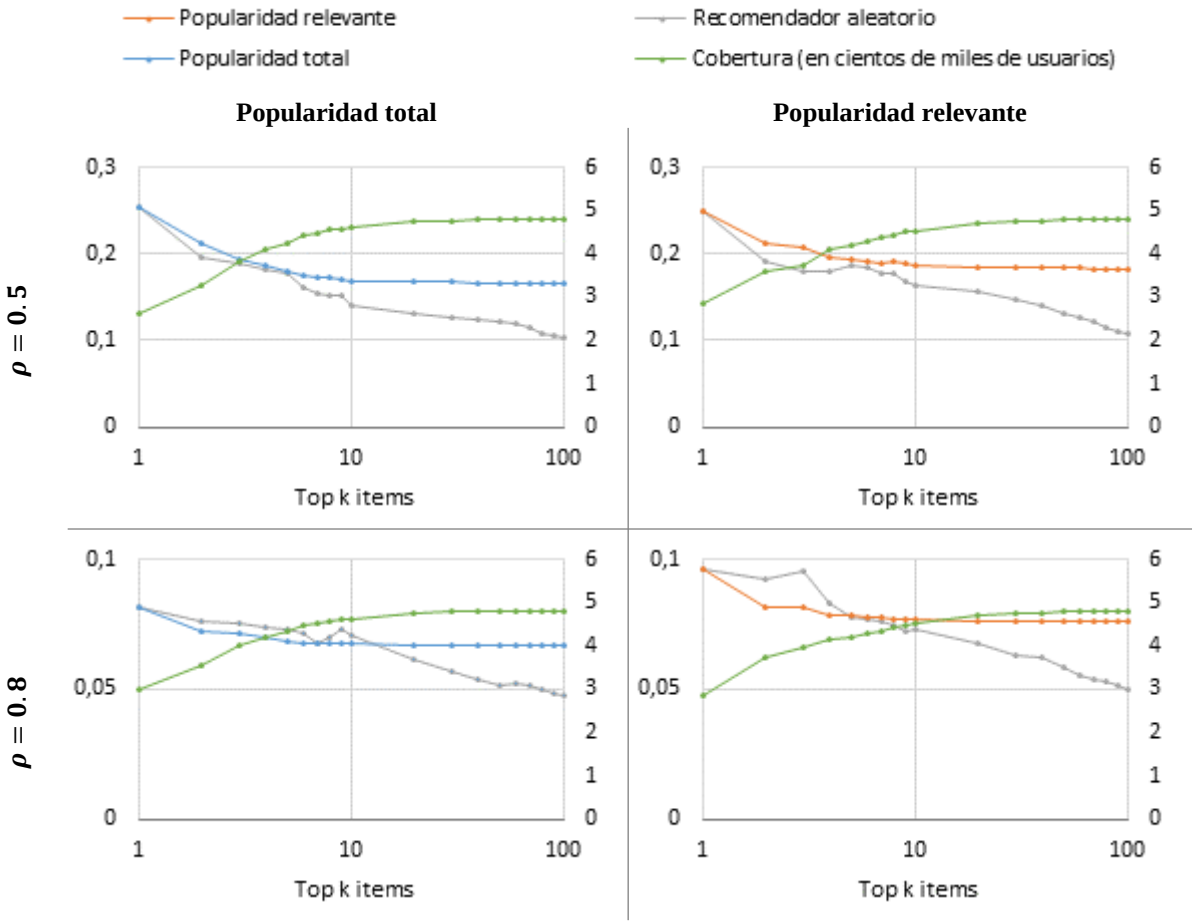

Figura 26. Precisión (observada) de los recomendadores al ejecutarlos sobre los datos de Netflix empleando para ello una partición temporal de tasa de entrenamiento 0.5 (fila superior) y 0.8 (fila inferior). En las gráficas, el eje x representa el número de ítems entre los que pueden elegir los recomendadores a la hora de evaluar, ordenados según su popularidad total (columna izquierda) y su popularidad relevante (columna derecha). Se incluye una curva que representa la evolución del número de usuarios a los que se les ha podido recomendar – aparece en color verde y sigue el eje y derecho medido en cientos de miles de usuarios.

Llama la atención el hecho de que la precisión de ambos tipos de popularidad disminuye en los primeros valores de k. En el caso del recomendador aleatorio este comportamiento es predecible, pues añadir ítems menos populares hace que dicho recomendador considere como candidatos ítems que, eventualmente, tendrán muy pocos votos en test. Popularidad, por el contrario, no debería verse afectada al aumentar los candidatos, pues siempre recomienda el más popular. Lo que ocurre es que aumenta la cobertura y, por tanto, usuarios que para un valor de k menor se habían descartado, en este caso sí que pueden obtener una recomendación. Su precisión se promedia con el resto de usuarios, por lo que en función de si es mayor o menor que la media la curva puede subir o bajar.

Respecto a la comparativa entre recomendadores, observamos que para una tasa de entrenamiento de 0.5, ambas popularidades presentan siempre una precisión observada superior a la del recomendador aleatorio. Sin embargo, al aumentar la tasa de entrenamiento a 0.8 se producen inversiones en las primeras posiciones, es decir, si nos limitamos a recomendar los 5 ítems más populares, es mejor – en términos de precisión observada – realizar una recomendación aleatoria que limitarse a recomendar el más popular.

Para entender qué ocurre cuando realizamos la partición con esta tasa de entrenamiento de 0.8 vamos a reducirnos al caso de popularidad relevante, pues en el caso de Netflix el sesgo de popularidad es muy grande y ambas popularidades son muy similares. Por ello, hasta el final de esta sección cuando hablemos de popularidad o de número de votos, nos estaremos refiriendo siempre a los relevantes.

En la Figura 27.a se muestra la evolución temporal del número de votos de los diez ítems más populares del conjunto de entrenamiento, desde el punto de inicio, es decir, desde la creación del primero de los diez ítems. El color indica el número de votos en test: cuanto más oscuro, más votos.

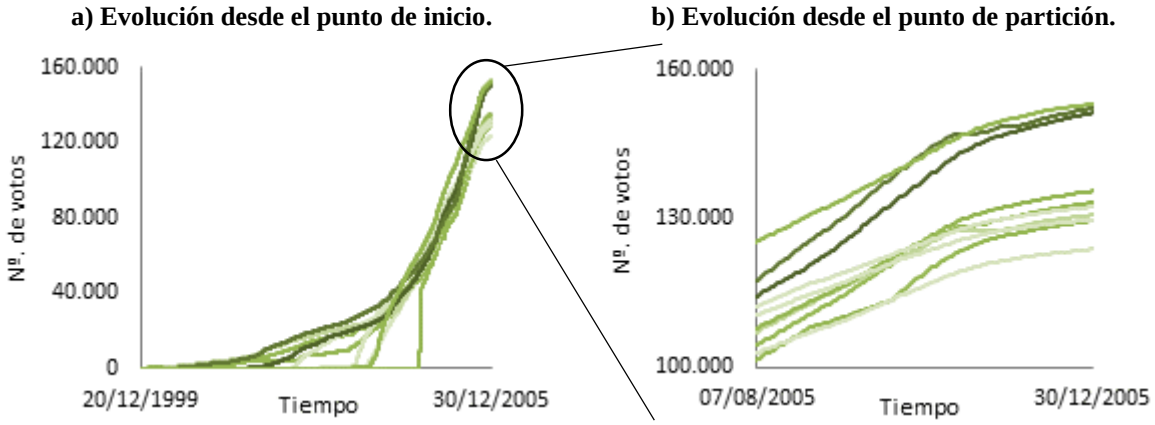

Figura 27. Evolución temporal del número de votos de los diez ítems más populares del conjunto de entrenamiento al realizar una partición temporal de tasa de entrenamiento 0.8 del conjunto de datos de Netflix. El color indica el número de votos en test: cuanto más oscuro, más votos. La gráfica de la izquierda (a) muestra toda la evolución desde la creación del primero de los ítems, mientras que la gráfica de la derecha (b) muestra la evolución desde el punto de inicio.

Observando la Figura 27.a se puede ver que las curvas presentan bastante sesgo en los momentos finales, por lo que el punto de partición se encuentra muy cercano al tiempo final, ya en el año 2005. Esto dificulta interpretar qué ítem tiene más popularidad y cual menos en ambos puntos. Para facilitar el análisis, en la Figura 27.b se muestra únicamente la evolución de los ítems a partir del punto de partición. De esta forma, se puede observar con mayor claridad

que el ítem con mayor popularidad en el conjunto de entrenamiento – el que se encuentra más arriba en el punto de partición – presenta menos votos en test que los dos siguientes. El que más votos tiene en test es precisamente el tercer ítem, por eso en la Figura 26 (fila inferior, columna derecha) cuando se emplea una partición de tasa de entrenamiento 0.8, el recomendador aleatorio presenta su máxima precisión al limitarse al top 3 de los ítems más populares. En dicho punto es cuanto más superior es a popularidad, pues este último está recomendando siempre el que menos votos relevantes tiene en test de los tres.

7.2.1.2 Twitter

La red social Twitter puede interpretarse como una red social en la que existen ítems sobre los que se realizan votos: los ítems son los tweets y los votos los retweets. Así, cuando un usuario retwittea un tweet se considera que está votándolo como relevante. En esta situación los votos son siempre positivos y ambos tipos de popularidad coinciden.

La particularidad de Twiter que resulta interesante en este estudio es que la velocidad a la que se propaga la información es muy alta y los tweets caducan – dejan de ser relevantes – en cuestión de horas. Así, generalmente no es efectivo recomendar un tweet que ha sido realizado hace más de dos días – por muy popular que haya llegado a ser – pues seguramente la información a la que haga referencia ya esté obsoleta.

Para el estudio hemos empleado el dataset de Twitter recabado por J. Sanz Cruzado que se puede obtener en la siguiente dirección: http://ir.ii.uam.es/datasets/twitter/twitter-temporal-1month.zip. Los tweets de dicho conjunto de datos pertenecen a un intervalo temporal de un mes – del 16 de junio al 16 de julio de 2015 – pero en nuestro caso interesa un periodo más corto para no penalizar a popularidad de forma ilógica, pues si el ítem más popular ha sido creado en los primeros quince días está claro que recomendarlo dos semanas más tarde va a ser una mala opción. De acuerdo con este razonamiento, nos hemos limitado a considerar los tweets y retweets del día 8 de julio de 2015. En concreto, los ítems se corresponden con los tweets creados durante las primeras 12 horas de dicho día, de forma que el conjunto de entrenamiento está formado por los retweets realizados en esas horas, mientras que el conjunto de test lo conforman los realizados en las 12 siguientes.

Las dimensiones de este conjunto se muestran en la Tabla 5, donde llama la atención la baja densidad del dataset. Este es un problema frecuente cuando se interpreta Twitter de esta forma, pues lo normal es que un tweet presente muy pocos retweets.

Estadísticas	
Nº. de usuarios	2.124
Nº. de ítems	2.309
Nº. de ratings	6.166
Densidad	0,12%

Tabla 5. Dimensiones del subconjunto de datos de Twitter correspondiente al día 8 de julio.

En la Figura 28 se muestra la evolución de la precisión observada de los recomendadores aleatorio y popularidad cuando se realiza la partición temporal de la forma descrita anteriormente. De forma análoga a las gráficas correspondientes para el conjunto de Netflix, el eje x representa el número de ítems entre los que pueden elegir los recomendadores, ordenados

según su popularidad. También se incluye la cobertura de los recomendadores, medida en miles de usuarios.

Observamos que reduciéndonos a los diez ítems más populares, emplear la recomendación aleatoria produce una precisión observada mayor que si recomendamos directamente el más popular.

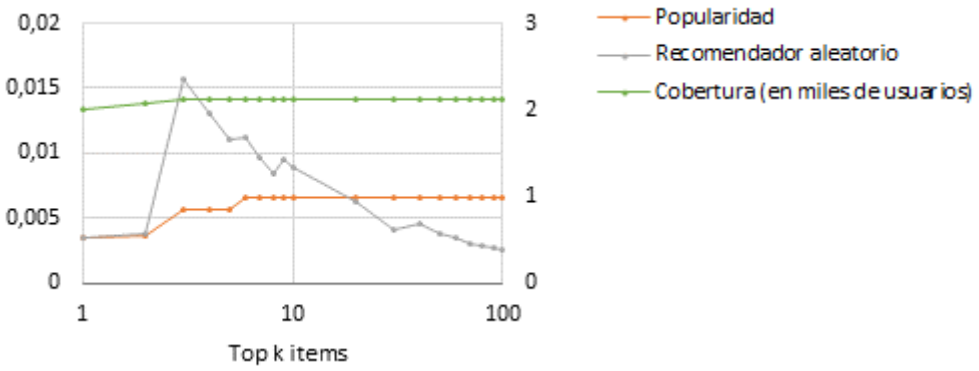

Figura 28. Evolución de la precisión observada de los recomendadores aleatorio y popularidad. El eje x representa el número de ítems entre los que pueden elegir los recomendadores, ordenados según su popularidad. También se incluye la cobertura de los recomendadores, medida en miles de usuarios.

En la Figura 29 se muestra la evolución temporal del número de votos de esos diez ítems. Cada curva representa un ítem, y el color informa acerca del número de votos en test: cuanto más oscuro, más votos. Constatamos en primer lugar el rápido aumento del número de votos tras crearse el tweet, ya que la pendiente es prácticamente vertical y no existe un periodo o fase inicial antes de dicho aumento. Sí que está presente, por el contrario, le etapa en la que la curva se estabiliza y el ítem recibe cada vez menos votos.

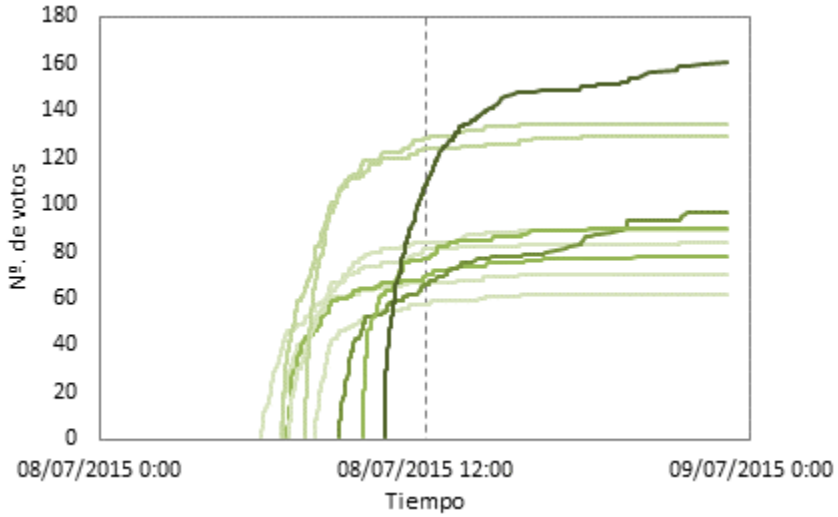

Figura 29. Evolución temporal del número de votos de los diez ítems más populares. Cada curva representa un ítem, y el color informa acerca del número de votos en test: cuanto más oscuro, más votos. La línea divisoria indica el punto de partición.

Respecto al crecimiento de los distintos ítems, observamos que los dos tweets más populares en entrenamiento están en fase de saturación y prácticamente no tienen votos en test. Estos ítems van a ser los recomendados por popularidad, motivo por el cual este recomendador presenta una precisión tan baja en comparación con el recomendador aleatorio. Entre los ítems

con mayor número de votos en test destaca el tercer ítem más popular en entrenamiento y el penúltimo. Este último nunca será recomendado por popularidad, pero si es tenido en cuenta por el recomendador aleatorio.

7.2.2 Datos sintéticos

En los dos ejemplos anteriores con datos reales, Netflix y Twitter, se pone de manifiesto que existe una cierta influencia del sesgo de la curva de popularidad de los ítems en la evaluación mediante una partición temporal. En esta sección estudiamos dicha influencia de forma más sistemática.

Para ello, tomamos los ratings del conjunto de Netflix y, mediante simulación, les asignamos una marca temporal de forma que la curva de popularidad resultante de cada ítem presente la forma de una función logística de un cierto sesgo. Por simplicidad, únicamente consideramos los votos relevantes de Netflix, y el punto de creación de cada ítem se elige de forma aleatoria.

La función de una curva logística viene dada por la siguiente expresión, donde C y β constantes y t la variable tiempo.

$$f(t) = \frac{e^{\beta t}}{C - 1 + e^{\beta t}}$$

En la literatura se ha visto que este tipo de funciones sirven para modelizar procesos de propagación en grafos completos (Newman 2010), como pueden ser las epidemias o los rumores. En dicha modelización, C es el número de nodos – generalmente personas – del grafo y β el número de contactos por unidad de tiempo. Es decir, β determina la velocidad de la curva, factor cuya influencia queremos analizar.

En nuestro caso, lo que se propaga por la red es la información acerca de un determinado ítem, por lo que es lógico considerar que los ítems más populares se propagan a mayor velocidad que los no populares. Para incluir esta consideración en la fórmula de la función logística multiplicamos el parámetro β por la fracción de usuarios que han votado el ítem. Así mismo, el valor de la constante C para cada ítem – el número de nodos a los que puede "propagarse" – es precisamente el número de usuarios que han votado dicho ítem. Por último, cabe señalar que la curva logística clásica devuelve el valor de la fracción de nodos a los que se ha propagado el ítem, por lo que para obtener el número absoluto multiplicamos dicho valor por la popularidad del ítem.

Con estas modificaciones, la función logística empleada en la simulación del crecimiento de cada ítem i presenta la siguiente forma, donde m denota el número de usuarios totales.

$$f(i,t) = |i| \frac{e^{\frac{\beta |i|}{m} t}}{|i| - 1 + e^{\frac{\beta |i|}{m} t}}$$

Nuestro objetivo es evaluar el comportamiento de los recomendadores en función del sesgo de las curvas, es decir, en función del valor de β. En concreto, variamos β entre 10 y 100 contactos por unidad de tiempo, porque valores fuera de ese rango producen curvas o muy planas o muy sesgadas que resultan poco realistas.

En la Figura 30 se muestra la forma de las curvas de los diez ítems más populares para los dos valores extremos de β (10 y 100) y un punto intermedio (50). Observamos que cuanto mayor es la velocidad de propagación – $\beta = 100$ – más sesgo presentan las curvas, con lo cual mayor

es la probabilidad de que todos sus votos se encuentren en el mismo conjunto, entrenamiento o test. Sin embargo, con pocos contactos por unidad de tiempo – $\beta = 10$ – la fase de crecimiento de los ítems dura más y la probabilidad de que haya un reparto equitativo de los votos entre entrenamiento y test es mayor.

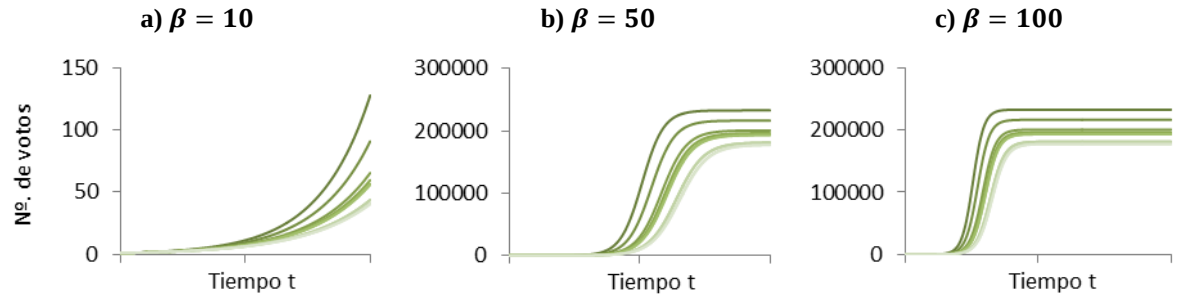

Figura 30. Evolución temporal del número de votos de los diez ítems con más votos, para unos sesgos de 10(a), 50(b) y 100 (b) contactos por unidad de tiempo. El final de la curva denota el punto de partición.

Una vez asignada la marca temporal de cada voto, realizamos una partición temporal que considera que el punto de partición está fijo y es siempre el mismo momento – en nuestro caso 1000 unidades de tiempo, el punto final de la Figura 30. Al dejar fijo el punto de partición pero aumentar el sesgo de las curvas de popularidad, se produce una disminución del número de votos en test que conlleva, a su vez, una bajada de la precisión del recomendador aleatorio, tal y como se observa en la Figura 31. En dicha figura se indica la evolución de las precisiones del recomendador aleatorio (limitado a los 1000 ítems más populares) y popularidad según aumenta el sesgo. Dado que todos los votos son positivos ambas popularidades coinciden.

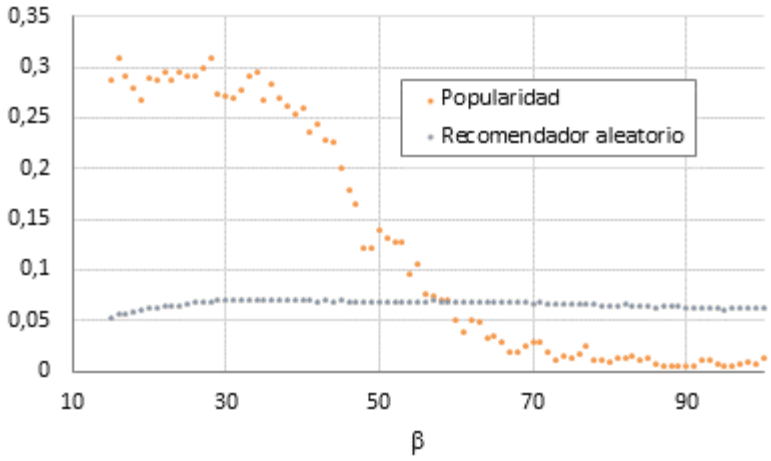

Figura 31. Evolución de la precisión de los recomendares popularidad y aleatorio en función de β, el sesgo de la curva logística.

Observamos que, tal y como esperábamos, para curvas con poco sesgo – β inferior a 50 – la popularidad obtiene mayor precisión que el recomendador aleatorio. Sin embargo, este comportamiento se invierte al considerar curvas cada vez más sesgadas. Como ya comentábamos anteriormente, lo que ocurre es que la correlación entre entrenamiento y test es positiva cuando las curvas son bastante planas, pero empieza a disminuir, incluso volverse negativa, según aumenta la pendiente de las curvas.

7.3 Influencia de la varianza

Otro fenómeno que depende del protocolo de partición y que puede llegar a afectar considerablemente al resultado de la evaluación es la diferencia de varianzas entre las distribuciones de los datos y la partición. Es un factor que, al igual que la partición temporal, también puede llegar a producir una correlación negativa entre el número de votos del conjunto de entrenamiento y del de test, aunque por motivos distintos a los de la partición temporal.

El mejor ejemplo en el que se aprecia con claridad este fenómeno es considerar que la distribución de popularidad de los datos es exacta, es decir, que todos los ítems tienen exactamente el mismo número de votos. En esta situación, la varianza de la partición, esto es, el hecho de que por cuestiones de azar los distintos ítems acaben teniendo una tasa de entrenamiento ligeramente diferente, adquiere un papel determinante. Así, el ítem que por azar consiga una mayor tasa de entrenamiento, será el que más número de votos tenga en entrenamiento – y será recomendado por popularidad – pero el que menos votos tiene en test y, con ello, su precisión será menor que la que se obtendría recomendando cualquier otro ítem. En este sentido, popularidad está recomendando justamente lo menos popular en test.

El fenómeno es bastante claro cuando se considera una distribución exacta, pero se ha observado que también puede producirse cuando existen pequeñas variaciones en el número de votos de unos ítems a otros. Por ejemplo, al emplear todos los datos obtenidos a partir del cuestionario realizado a los usuarios de Crowdflower, se observa que popularidad total obtiene una precisión observada ligeramente inferior a la del recomendador aleatorio (Figura 24 de la sección 6.3). Esto ocurre porque el número de votos por ítem depende de a cuantos usuarios les haya sido asignado para votarlo, proceso que es aleatorio. Por ello, la distribución de popularidad es muy uniforme, tal y como se observa en la Figura 32.

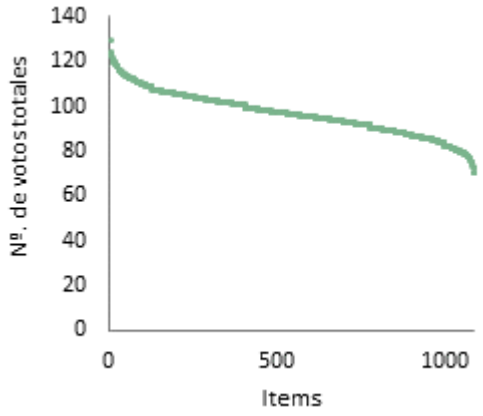

Figura 32. Distribución de popularidad de todo el conjunto de preferencias obtenidas a partir del cuestionario a los usuarios de Crowdflower.

Al presentar los ítems una popularidad tan semejante, el ítem que por azar consiga un mayor número de votos en entrenamiento generalmente tendrá menos votos en test y, con ello, su precisión será menor que la de otros ítems.

Más allá de estos ejemplos, hemos observado que este efecto cobra mayor influencia cuanta menor densidad existe – ya que la varianza de los datos disminuye con ella – y cuanto mayor es la proporción de ítems frente a usuarios. Sin embargo, un análisis más profundo se escapa del alcance de este trabajo y constituye una de las líneas para desarrollar en un trabajo futuro.

8. Ampliación de perspectiva

Aunque el presente trabajo se centra en estudiar específicamente la recomendación por popularidad, ya en el capítulo 2 avanzábamos la relación que esta recomendación presenta con otros recomendadores más complejos y utilizados en la literatura, como la factorización de matrices o el filtrado por vecinos próximos. Por eso, resulta relevante observar cómo las observaciones sobre la popularidad pura pueden extrapolarse a otros recomendadores comunes.

Concretamente, tomaremos el experimento realizado con los usuarios de Crowdflower, pues es el único que permite considerar las precisiones real y observada, y ampliaremos los estudios realizados con la popularidad al resto de recomendadores. Esta es una investigación todavía en curso en la que cabe realizar muchas consideraciones, pero en este trabajo nos limitamos a constatar los resultados, dejando un análisis más profundo de los mismos para futuros proyectos.

Además de considerar otros recomendadores, resulta interesante contrastar si los fenómenos que observamos al considerar como métrica la precisión del primer ítem recomendado ($P@1$) se mantienen cuando empleamos otras métricas que consideran un ránking de más elementos. En la segunda sección de este capítulo realizamos la comparativa de los recomendadores empleando otras métricas de ránking, como recall, nDCG o la propia precisión extendida a un número mayor de posiciones.

8.1 Otros recomendadores

En la Figura 33 se muestra la precisión observada y real en la primera posición de la recomendación producida por diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower. La fila superior se corresponde con una partición aleatoria cuya tasa de entrenamiento ρ es 0.5 y la inferior con una tasa de entrenamiento 0.8.

Los recomendadores coinciden con los empleados en la comparativa del capítulo 2, incluidas sus configuraciones de parámetros, salvo en el caso de los kNN que debido a las pequeñas dimensiones del dataset hemos tomado todos los usuarios como vecinos. Recordamos que la similitud empleada por dichos algoritmos era la similitud Jaccard. Respecto al algoritmo de factorización de matrices, se toman 50 factores, $\lambda = 0.1$, $\alpha = 1.0$ y se realizan 20 iteraciones.

Llama la atención, en primer lugar, que para ambas tasas de entrenamiento en la precisión observada las popularidades superan al resto de algoritmos, siendo el kNN ub sin normalizar el que más se aproxima a ellas. También cabe destacar la baja precisión de kNN ib sin normalizar, pues se encuentra prácticamente al mismo nivel que los normalizados. Respecto al rating promedio y los kNN normalizados, su comportamiento el similar al que observamos en el capítulo 2 al ejecutar los algoritmos sobre los conjuntos de datos de MovieLens, Netflix y Last.fm. En un análisis más detallado sería conveniente intentar modificar los parámetros de los distintos algoritmos para conseguir que, al menos los kNN sin normalizar y factorización de matrices, superen a las popularidades. Sin embargo, en los tanteos realizados a este respecto con los kNN – probando con distintos tamaños de vecinos y similitudes – no hemos encontrado una configuración que supere a las popularidades.

Respecto a la comparativa entre la precisión real y la observada, cabe destacar varios aspectos. En primer lugar, y para ambas tasas de entrenamiento, se produce un aumento significativo del rating promedio respecto al resto de algoritmos, lo que los deja a todos prácticamente al mismo nivel.

Cuando la tasa de entrenamiento es 0.5 en general el orden entre los recomendadores se mantiene, siendo las popularidades y kNN ub los que superan al resto. Sin embargo, con una tasa de entrenamiento de 0.8 se produce la inversión con el recomendador aleatorio, que afecta también a kNN ub aunque en menor medida que a las popularidades. Es decir, en esta situación se tiene que la recomendación mediante vecinos próximos basada en usuario, muy frecuentemente utilizada en la literatura, presenta una precisión real inferior a la recomendación aleatoria. Un fenómeno que no sabemos si únicamente conocemos la precisión observada, donde la ordenación es la contraria.

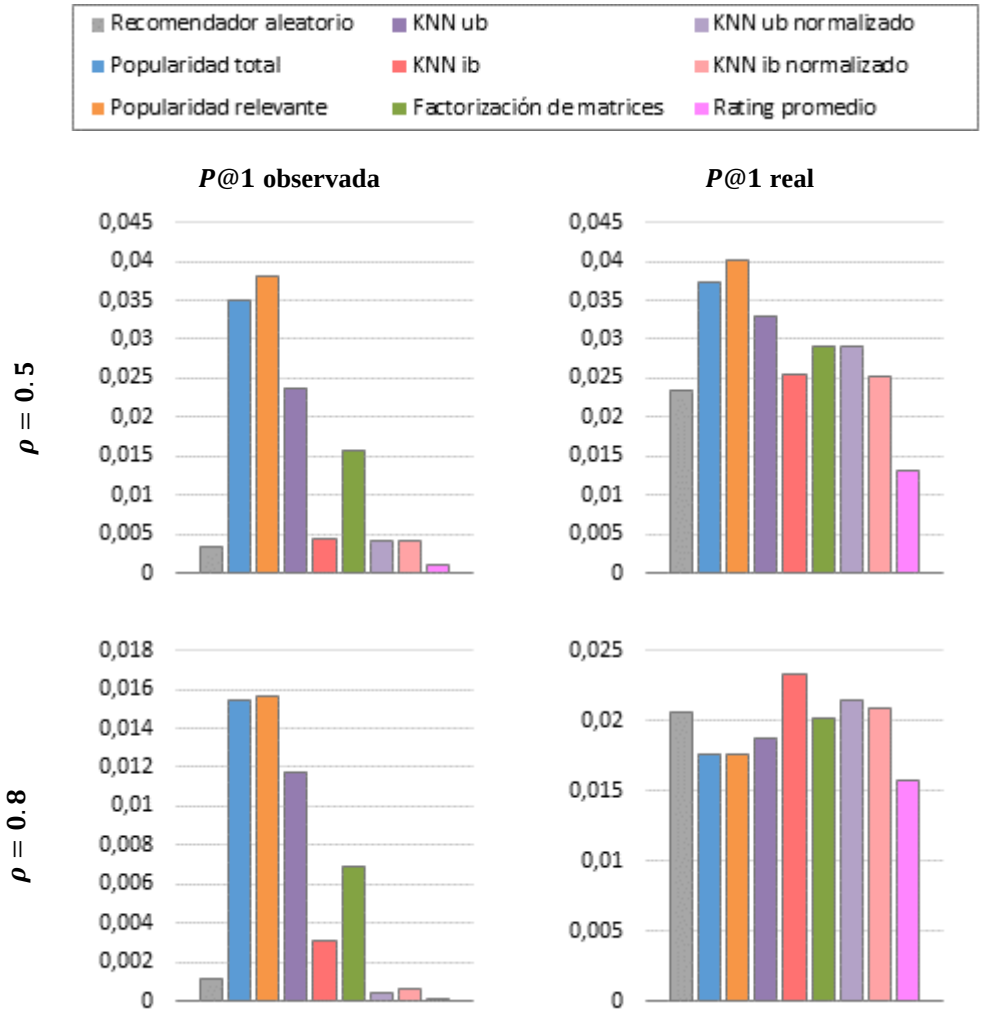

Figura 33. Precisión observada (columna izquierda) y real (columna derecha) en la primera posición de la recomendación producida por diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower. La fila superior se corresponde con una partición aleatoria cuya tasa de entrenamiento ρ es 0.5 y la inferior con una tasa de entrenamiento 0.8.

Resulta relevante recordar las pequeñas dimensiones del conjunto, que consta de unos 1000 usuarios y 1000 ítems (los datos concretos pueden consultarse en la Tabla 4). Es decir, que en las precisiones observadas la diferencia entre los recomendadores con menor y mayor precisión

es de en torno a 15 usuarios. Esta diferencia se reduce ligeramente a 10 usuarios cuando consideramos la precisión real.

Estas dimensiones también podrían causar, junto con la asignación aleatoria de votos, que existan pocas agrupaciones de usuarios e ítems, lo cual justificaría la baja precisión observada en los kNN y factorización de matrices.

Probamos ahora a hibridar los algoritmos anteriores con popularidad. Así, en la Figura 34 se muestra la evolución de dichos algoritmos cuando limitamos sus recomendaciones a los k ítems más populares.

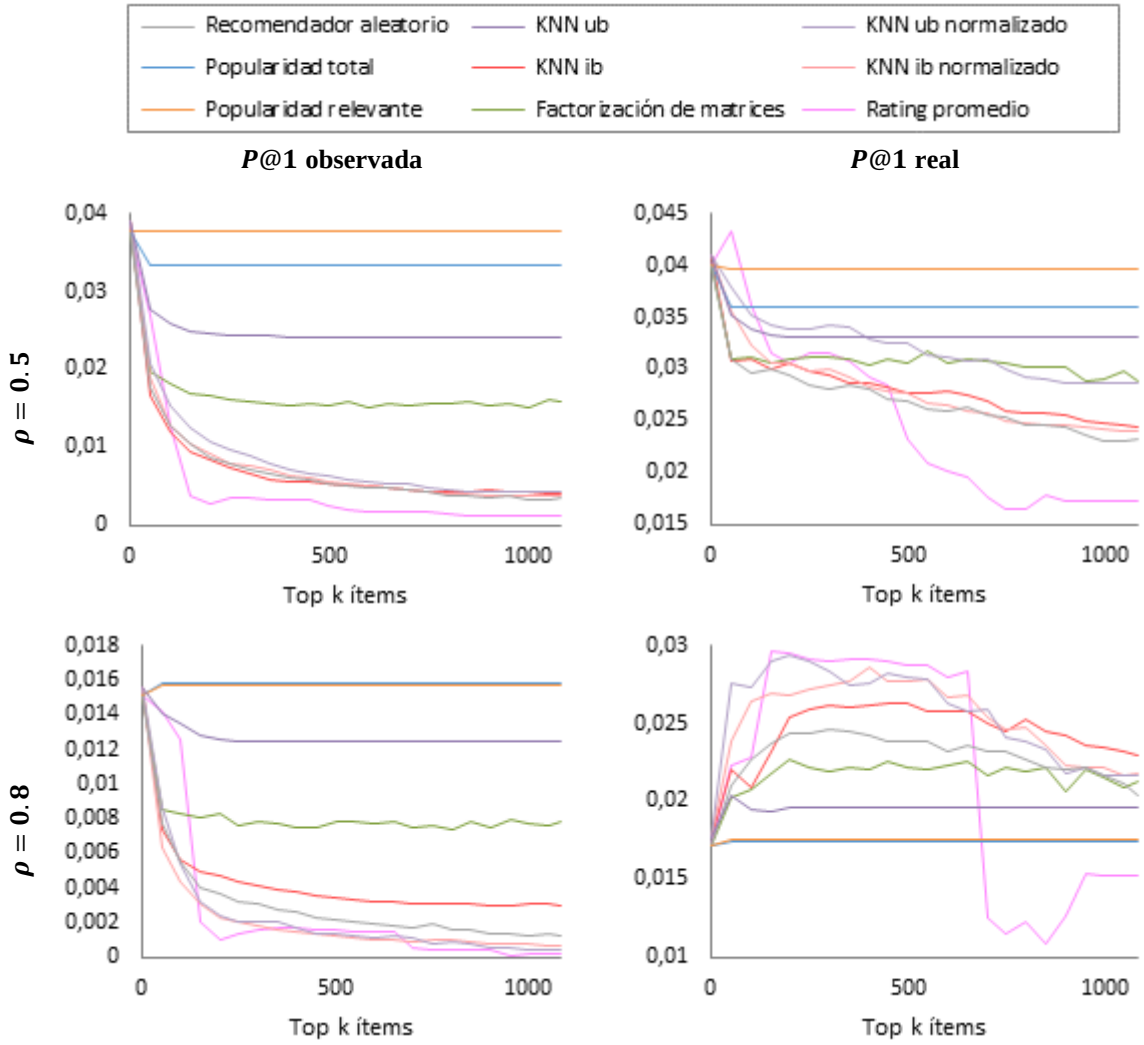

Figura 34. Evolución de la precisión observada (columna izquierda) y real (columna derecha) de diversos recomendadores al limitar sus recomendaciones a los k ítems más populares. La fila superior se corresponde con una partición aleatoria cuya tasa de entrenamiento ρ es 0.5 y la inferior con una tasa de entrenamiento 0.8. El eje x de las gráficas se incrementa de 50 en 50.

No se detectan comportamientos destacables en cuanto a la precisión observada, pues para ambas tasas de entrenamiento los algoritmos tienden a estabilizarse pasado el top 100. Sin embargo, en la precisión real de la partición con rho 0.8 llama la atención que el rating promedio presenta la precisión más alta hasta introducir los ítems menos populares, cuando desciende bruscamente. Con una tasa de entrenamiento menor, sin embargo, este comportamiento únicamente se observa dentro de los 50 ítems más populares.

El motivo es que aquellos ítems poco populares cuyos votos sean todos relevantes obtienen un rating promedio muy elevado y distorsionan la recomendación, sesgándola en contra de popularidad. Los kNN normalizados siguen el comportamiento del rating promedio al considerar los primeros ítems populares, pero su caída es menor cuando interviene la cola de la distribución de popularidad y resisten quedándose al mismo nivel que el recomendador aleatorio.

También en esta misma gráfica (precisión real y tasa de entrenamiento 0.8) llama la atención que factorización de matrices no parece funcionar bien cuando se limita a los ítems más populares, pues se encuentra por debajo de la recomendación aleatoria hasta prácticamente el final. Cabría considerar otras configuraciones de sus parámetros para comprobar si este comportamiento es sistemático o se trata simplemente de que no se ha encontrado el ajuste óptimo. Dicho análisis, sin embargo, se escapa del alcance de este trabajo, en donde nos limitamos a constatar que el mismo algoritmo que en los conjuntos de MovieLens, Netflix y Last.fm obtenía una precisión superior al resto, en este dataset se encuentra al nivel del recomendador aleatorio.

Un último estudio para contextualizar la precisión obtenida por los algoritmos consiste en enfrentar el número de veces que cada ítem es recomendado con su popularidad, de forma similar a como hacíamos en el capítulo 2. Así en la Figura 35 se muestra dicha comparativa con una tasa de entrenamiento 0.8 (con 0.5 es prácticamente igual). A diferencia del estudio realizado en el capítulo 2, en este caso los recomendadores realizar únicamente una recomendación (es un corte en la primera posición), y se incorpora a la comparativa con la popularidad de cada ítem – fila superior – la comparativa con su rating promedio – fila inferior.

En primer lugar cabe señalar que no se observan diferencias destacables entre los comportamientos de los recomendadores para las dos tasas de entrenamiento. Vemos que en ambos casos el algoritmo que más se sesga hacia los ítems populares es claramente kNN ub, lo cual concuerda con el hecho de que al considerar su precisión (Figura 33) es el que más se asemeja al comportamiento de las popularidades, tanto en real como en observada.

Observamos más claramente en estas figuras el comportamiento opuesto entre la popularidad y el rating promedio, pues el ítem más recomendado por este último es precisamente el menos popular. Respecto a los kNN normalizados, se vuelve a constatar su tendencia a recomendar los ítems con mayor rating promedio, aunque se observa que evitan sesgarse hacía los extremos poco populares, lo que concuerda con el hecho de que en la Figura 34 su evolución al incluir los ítems menos populares resista y no baje tan bruscamente como el rating promedio.

Por último, llama la atención que factorización de matrices, aunque en general presenta una cierta tendencia hacia lo popular, el ítem que más recomienda es precisamente el que de menor popularidad, que coincide con el de mayor rating promedio. Este comportamiento es completamente inesperado y analizarlo forma parte del trabajo futuro de este proyecto.

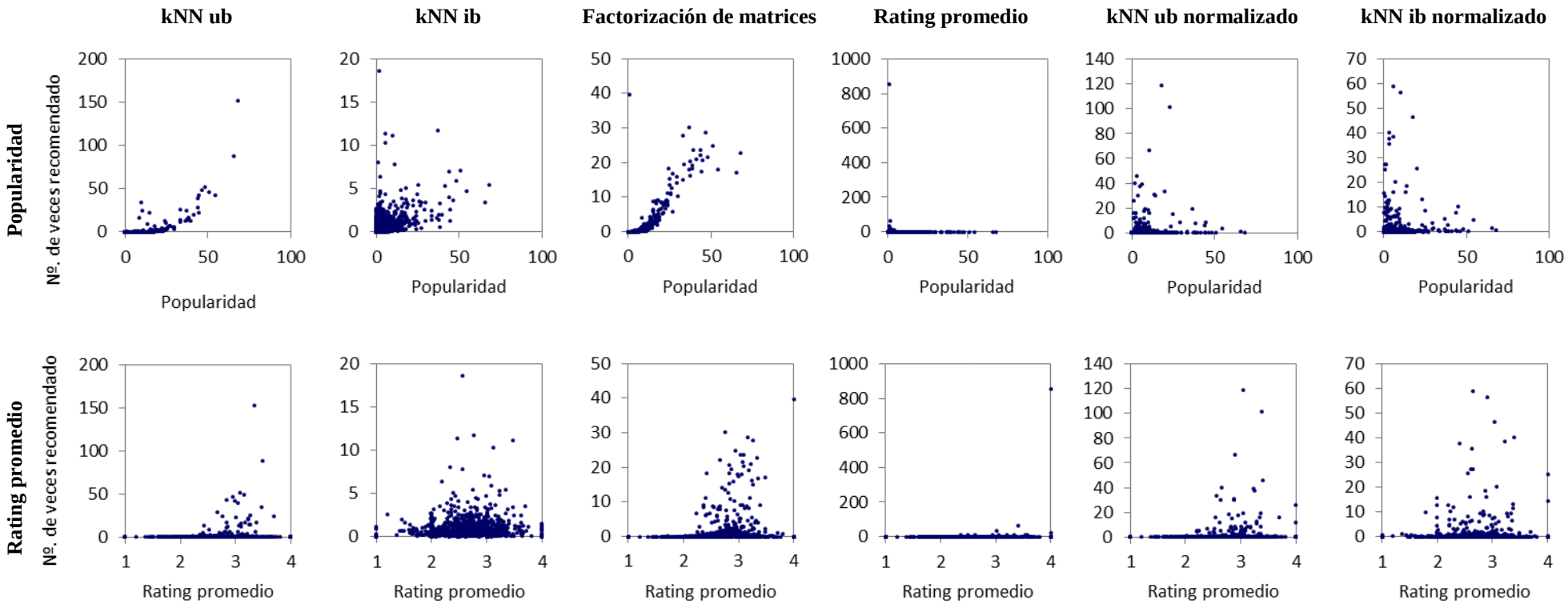

Figura 35. Número de veces que se recomienda cada ítem frente a la popularidad (relevante) – fila superior – que presenta dicho ítem y frente a su rating promedio – fila inferior – para diversos recomendadores. Los ránkings son únicamente de 1 elemento y la tasa de entrenamiento es 0.8.

8.2 Otras métricas

El estudio de la sección anterior se basa en la comparativa de distintos algoritmos empleando para ello la métrica $P@1$. En esta sección extendemos dicho análisis a otras métricas más frecuentes – que consideran un ránking con más elementos – con el objetivo de constatar si se mantiene la comparativa. En concreto se han empleado las métricas precisión, recall, MRR y nDCG sobre los diez primeros elementos de las recomendaciones.

En las Figuras 36 y 37 – tasas de entrenamiento 0.5 y 0.8, respectivamente – observamos la comparativa entre $P@1$ y $P@10$, que se muestran juntas para poder contrastar cómo afecta el aumentar el tamaño del ránking.

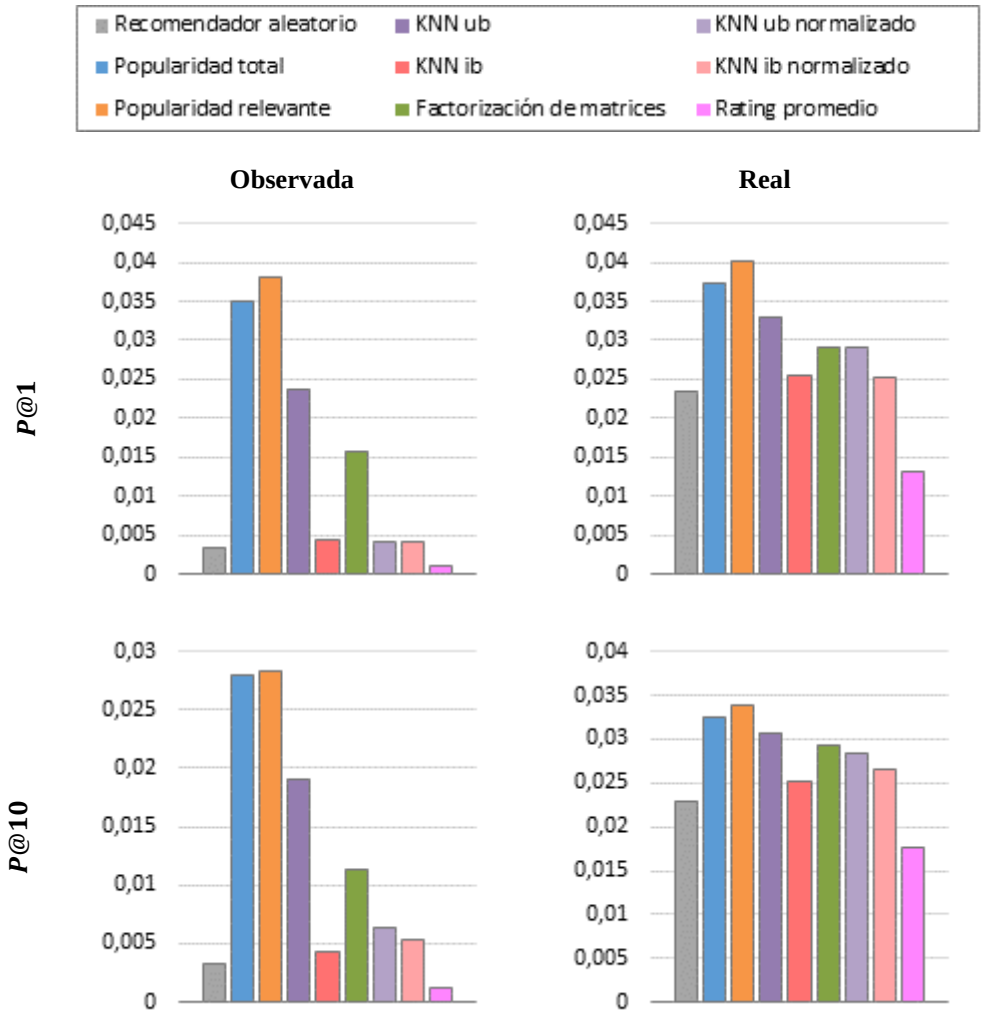

Figura 36. Precisión observada (columna izquierda) y real (columna derecha) de $P@1$ y $P@10$ sobre diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower con tasa de entrenamiento ρ de 0.5.

Vemos que para ambas tasas de entrenamiento el comportamiento de ambas métricas es prácticamente idéntico. En particular, cabe resaltar que para $P@10$ también se produce una inversión del recomendador aleatorio con las popularidades y kNN ub al considerar la precisión real.

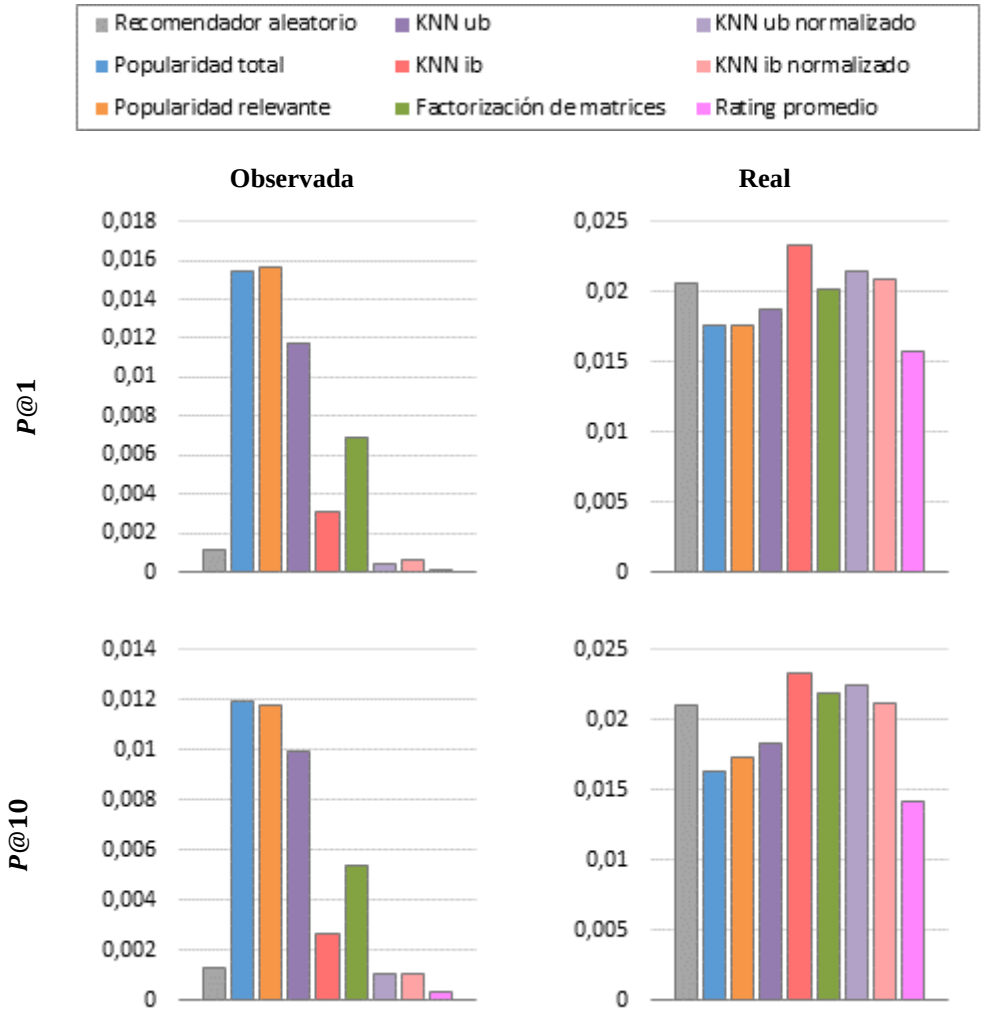

Figura 37. Precisión observada (columna izquierda) y real (columna derecha) de $P@1$ y $P@10$ sobre diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower con tasa de entrenamiento ρ de 0.8.

Análogamente, en las Figuras 38 y 39 del Anexo 2 podemos observar los valores de las otras métricas (MRR, recall y nDCG) junto con $P@10$.

Vemos que con una tasa de entrenamiento de 0.5 el comportamiento entre las tres métricas es prácticamente idéntico y muy similar al de $P@1$ y $P@10$. Con $\rho = 0.8$, sin embargo, las métricas presentan diferencias entre sí. Con MRR se produce la inversión en el valor real del recomendador aleatorio con las popularidades y kNN ub, al igual que ocurría con $P@1$ y $P@10$. Sin embargo, en recall y nDCG no ocurre esto y ambos valores – real y observado – coinciden en colocar a las popularidades por encima del recomendador aleatorio.

Cabe señalar en este punto que precisión y MRR son métricas volumétricas – miden la cantidad de acierto global – por lo que los usuarios más activos tienen más peso, al ser mayor la probabilidad de acertar con ellos. Recall y nDCG, por el contrario, dividen por el número de ítems relevantes del usuario, por lo que restan peso a los usuarios más activos equiparándolos al resto. Estas dos últimas métricas se emplean, por tanto, para igualar la satisfacción por usuario.

El hecho de que con las métricas que igualan por usuario no se produzca la inversión nos sugiere que el problema se encuentra en recomendar lo popular a los usuarios más activos, y que es en ellos precisamente donde es peor – en términos de la relevancia real – que la recomendación aleatoria.

En general, y salvo estas excepciones, la conclusión es que se mantiene el comportamiento que observábamos al emplear la precisión de la primera posición del ránking. Así, si nos fijamos en las Figuras 40 y 41 del Anexo 2, que enfrentan el número de veces que se recomienda cada ítem contra su popularidad y su rating promedio, cuando se emplean ránkings de diez ítems, vemos que el comportamiento es prácticamente el mismo que el que observábamos en la Figura 35 en la que únicamente se utilizaba la primera posición.

9. Conclusiones

El presente trabajo aborda la pregunta de hasta qué punto es la popularidad efectiva o un ingrediente deseable en un recomendador, en qué medida y bajo qué circunstancias. El trabajo busca arrojar luz sobre esta pregunta mediante el desarrollo de formulaciones y análisis teóricos, el contraste empírico de dichas formulaciones teóricas, la configuración de simulaciones basadas en casos particulares de la formulación, y la realización y posterior análisis de experimentos con usuarios reales. Recapitulamos a continuación los aspectos más relevantes del trabajo realizado, y planteamos las múltiples líneas de trabajo futuro que se abren a partir del trabajo desarrollado hasta aquí.

9.1 Resumen y contribuciones

Los datos que se utilizan en la ejecución y evaluación de algoritmos de recomendación tienen fuertes sesgos en la distribución de las observaciones, esto es, en la distribución de popularidad de los ítems. Recientemente (Cremonesi et al 2010) se ha constatado la gran influencia que dichos sesgos tienen tanto en la recomendación – una gran mayoría de los algoritmos del estado del arte tienden a recomendar los ítems más populares – como en la evaluación – cuanto mayor es el sesgo más efectivo es recomendar lo popular. Estos sesgos de popularidad no se dan únicamente en los datasets con los que se hacen experimentos de laboratorio, también los sufren las aplicaciones reales, que trabajan con datos que ellas mismas recaban y están sujetas al encuentro espontáneo (o provocado por el mismo sistema de recomendación) entre usuarios e ítems. Es relevante, por tanto, entender en qué medida y bajo qué circunstancias la recomendación por popularidad es una técnica eficaz o no, pues de esta comprensión se derivan conclusiones acerca de la efectividad de otros muchos algoritmos que se encuentran influenciados por la distribución de popularidad.

El presente trabajo ha consistido, en primer lugar, en desarrollar una formulación teórica de la efectividad de un recomendador, distinguiendo entre la efectividad observada (la que habitualmente se obtiene en los experimentos que se realizan en el área) y la real. A partir de esta formulación ha sido posible estudiar la influencia en dicha efectividad de distintos aspectos, como el tipo de partición, la distribución de popularidad, la de descubrimiento, los gustos del usuario o su comportamiento a la hora de decidir sobre lo que votar o no.

Junto con el estudio analítico, llevamos a cabo un contraste empírico de hipótesis, empleando para ello conjuntos de prueba provenientes tanto de datos reales como de simulaciones. A este respecto destaca la realización de un experimento con usuarios reales de una plataforma de crowdsourcing. Lo novedoso de dicho experimento es que se ha realizado en condiciones de ausencia de sesgos – como los de descubrimiento – y, por tanto, lo que se obtiene es únicamente fruto de los gustos de los usuarios. Al mismo tiempo, las preguntas explícitas que se hacen a los usuarios permiten obtener una aproximación a los datos que normalmente se obtendrían en los procedimientos de recolección habituales (con todos sus sesgos), y los que habitualmente no serían observables (que en nuestra muestra obtenemos de forma expresa), lo que nos permite reproducir y analizar la diferencia entre resultados observables y reales.

La obtención expresa de datos de usuarios revela una comparación de las efectividades medida y real que contradice lo observable con los conjuntos de datos habitualmente disponibles, y arroja luz sobre las posibles causas de esta discrepancia, permitiendo un análisis de las mismas y complementando el análisis teórico

A nivel específico, las principales conclusiones derivadas del análisis teórico y del posterior contraste empírico se resumen en las siguientes:

- La popularidad relevante es más robusta y resistente que la popularidad total como criterio de recomendación. Así, salvo en ejemplos muy concretos y excepcionales, la primera se encuentra por encima de la segunda. El motivo es que la efectividad de la popularidad total depende en gran medida de la proporción de votos relevantes que se realicen, de forma que si se descubre más o se vota más lo no relevante, dicha efectividad descenderá por debajo de la del recomendador aleatorio, pues los ítems con más votos serán precisamente los que menos votos relevantes presenten. La popularidad relevante, sin embargo, resiste este tipo de fenómenos y para vencer su efectividad es necesario recrear situaciones más excepcionales. Por este motivo, para las siguientes conclusiones nos vamos a limitar a tratar el caso de la popularidad relevante.
- Si el descubrimiento de los ítems es neutro – no depende ni de la relevancia ni del ítem en concreto – entonces la popularidad relevante presenta un ránking óptimo en términos tanto de precisión observada como de precisión real. Por supuesto, cuando hablamos de ránking óptimo en este sentido consideramos únicamente los recomendadores no personalizados (mismo ránking para todos los usuarios).
- Si el descubrimiento de los ítems tiene relación tanto con lo relevantes que son como con las características concretas de cada ítem – $p(seen|\ rel, i)$ –, entonces cualquier situación es posible: la popularidad relevante puede presentar una efectividad alta, neutra o baja (es decir, mejor, equivalente o peor que la recomendación aleatoria), tanto medida como real.

  Uno de los fenómenos más llamativos que pueden producirse, y que de hecho hemos recreado mediante el experimento con usuarios reales de la plataforma Crowdflower, es que se produzca una discrepancia entre la efectividad medida y la real al comparar dos recomendadores (en este caso la recomendación popular y la aleatoria). En un experimento de evaluación offline típico únicamente se dispone de la efectividad medida, con la que se asume se está aproximando la real, pero si dicha asunción es falsa – como ocurre en Crowdflower – las conclusiones del experimento son no únicamente erróneas, sino contrarias a las reales: podemos estar calificando como mejor un algoritmo que en realidad es peor.

  En general, que se produzcan unas situaciones u otras depende de las distribuciones concretas de descubrimiento y relevancia, y no es posible realizar afirmaciones atendiendo únicamente a las dependencias entre ambas. En estas condiciones, además, ordenar por la relevancia total de los ítems no tiene por qué producir un buen ránking en términos de efectividad real. Para ello, es necesario ordenar en su lugar por el siguiente cociente, que no es sencillo de interpretar a no ser que se realicen ciertas simplificaciones:

$$\frac{p(rel|i)\left(1-\rho\ p(rate|seen,rel)\ p(seen|\ rel,i)\right)}{1-\rho\ p(rate|seen,i)\ p(seen|\ i)}$$

En cualquier otra situación – descubrimiento neutro o únicamente dependiente de un factor (el ítem o la relevancia) – ordenar por relevancia produce un ránking óptimo en términos de efectividad real. Estas situaciones, sin embargo, requieren condiciones de independencia que no suelen darse en la realidad.

Por otro lado, el hecho de que ordenar por relevancia sea óptimo no asegura una alta efectividad de la popularidad relevante, pues es necesario que dicha popularidad ordene por relevancia. Para ello, las distribuciones de relevancia y descubrimiento deben preservar el mismo orden. En caso contrario, nuevamente cualquier situación es posible.

- La tasa de entrenamiento puede alterar la comparativa en cuanto a la efectividad real, haciendo que la popularidad resulte mejor que el recomendador aleatorio cuando dicha tasa presenta un valor lo suficientemente pequeño y peor cuando es suficientemente grande. Este comportamiento ha sido constatado empíricamente con los datos obtenidos de los usuarios en Crowdflower.

  Intuitivamente, lo que ocurre es que los ítems más conocidos saturan su relevancia – llegan a todos los usuarios a los que potencialmente les gustan – por lo que aquellos que no los conocen es porque generalmente no les son relevantes. De esta forma, al ampliar la tasa de entrenamiento y eliminar del conjunto de test – para pasarlos a entrenamiento – votos de usuarios que conocen el ítem, quedan únicamente en dicho conjunto los votos de los usuarios que no lo conocían que suelen considerar no relevantes a estos ítems tan populares.

  En este sentido, cuando la tasa de entrenamiento es elevada, los mejores ítems para recomendar son aquellos que han sido descubiertos por muchos usuarios a los que no gustan – y por tanto son excluidos de la recomendación a dichos usuarios – pero no son prácticamente conocidos por aquellos usuarios a los que sí gustan – y por tanto al recomendarlos serán un acierto.

- Las particiones temporales tienen el potencial de invertir la correlación de los ránkings de popularidad en los conjuntos de entrenamiento y test, haciendo que la popularidad presente una efectividad peor que la del recomendador aleatorio. Este fenómeno es más probable cuanto mayor sea la velocidad a la que los ítems adquieren votos y caducan, por lo que obviamente lo hemos observado en escenarios en que dicha velocidad es muy elevada, como Twitter. Sin embargo, también se da en Netflix cuando se toman particiones con una tasa de entrenamiento lo suficientemente grande.

9.2 Trabajo futuro

A lo largo del trabajo hemos señalado varias posibilidades y variables de interés para un posible trabajo futuro. En esta sección exponemos de forma sintetizada todas ellas y algunas más que no hemos mencionado explícitamente hasta ahora.

- Realizar un análisis equivalente al realizado con la popularidad pero con el rating promedio como función de ránking, que como hemos visto en el experimento de la sección 8, parecería ser más efectivo que la popularidad relevante en términos de efectividad real. Recordamos de la sección 2.2.2 que el rating promedio puede ser interpretado como una noción alternativa de la popularidad, y representa lo relevante que es el producto para los usuarios que lo han votado. Recomendar mediante este criterio consiste, por tanto, en

ordenar mediante el cociente entre el número de votos relevantes y el de votos totales. En nuestro marco teórico, este cociente corresponde con la probabilidad $p(rel|rate, i)$ – mientras que la popularidad relevante era $p(rel, rate|i)$.

- Extender el estudio empírico a otras métricas y recomendadores. A este respecto hemos realizado algunos avances, como los mostrados en el capítulo 8, pero queda pendiente realizar una tarea de ajuste para configurar adecuadamente los parámetros de los distintos algoritmos. Se trata, por tanto, de un trabajo en curso en el que aparecen nuevos fenómenos por explicar.
- Estudiar la influencia de los sesgos en el comportamiento de los usuarios en la efectividad de los algoritmos, es decir, el hecho de que unos sean más activos – introduzcan más votos – que otros puede influir en la efectividad de unos algoritmos (en particular el elemento de la popularidad) frente a otros.
- Modelizar la distribución de $p(rate|u, i)$ de forma que permita un ajuste preciso. Como se indica en la sección 5.1.2, esta expresión no permite un desarrollo cerrado por lo que para el análisis teórico llevado a cabo en este trabajo asumimos la independencia respecto al usuario. Sin embargo, sería posible aproximarla mediante una expresión parametrizada cuyos parámetros se ajusten a partir del conjunto de datos.
- En relación a la influencia del protocolo de partición se abren varias posibles vías para futuros estudios:
  - Estudiar casos donde cada ítem i tenga una tasa de entrenamiento distinta ρ_i. Más concretamente, probar distintas distribuciones de los ρ_i y analizar cómo influyen en el resultado de la evaluación de los recomendadores.
  - Continuar con la línea de lo expuesto en la sección 7.3, analizando más en detalle cómo influye la diferencia de varianzas de la partición y los datos en que se produzcan inversiones en las distribuciones de popularidad de entrenamiento y test. Estudiar, así mismo, de qué otros aspectos pueden depender dichas inversiones, pues hemos visto que la densidad de ratings o la proporción entre número de ítems frente a número usuarios también parecen tener una cierta influencia en los resultados.
  - Considerar otros tipos comunes de partición: fija por usuario, por ítem, *leave one out*, etc.
- Modelizar el descubrimiento como el resultado de la comunicación en una red social y analizar la influencia de dicha comunicación (si se habla de lo más relevante o no, por ejemplo), conectando el presente trabajo con el realizado en (Cañamares & Castells 2014).
- Una de las asunciones de nuestro modelo es que la relevancia, esto es, los gustos de los usuarios, permanece constante a lo largo de todo el proceso. Sin embargo, en la realidad nuestros gustos evolucionan en función de diversas influencias, como los gustos de nuestros amigos, la publicidad, las críticas, etc. Por tanto, un posible trabajo futuro sería considerar una relevancia variable y estudiar la evolución y formación de dicha relevancia en función de distintos parámetros.
- En este trabajo consideramos un escenario de recomendación en el que no se permiten recomendaciones repetidas, es decir, no se recomienda aquello que el usuario ya ha votado. Sin embargo, cabría generalizar el análisis relajando esta suposición, admitiendo la

posibilidad de recomendar ítems que los usuarios ya han consumido, por ejemplo, a modo de recordatorio de que dicho ítem les gusta. En dicho caso, en lugar de relevancia, emplearíamos para evaluar el acierto de una recomendación una cierta función de utilidad que pondere la relevancia por factores como el número de veces que el usuario ha consumido el ítem (k) o el tiempo que hace desde que lo hizo por última vez (t).

$$U(u,i) = rel(u,i) \cdot f(u,i,k,t)$$

En esta perspectiva, nuestro trabajo hasta aquí habría considerado un caso particular donde el término de ponderación es una función escalón que vale 1 si el ítem no ha votado el ítem y 0 en caso contrario.

$$f(u,i,k,t) = f(x) = \begin{cases} 1, si\ rate(u,i) = 0 \\ 0, en\ otro\ caso \end{cases}$$

Referencias

G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), June 2005, pp. 734-749.

P. Adamopoulos, A. Tuzhilin and P. Mountanos. Measuring the Concentration Reinforcement Bias of Recommender Systems. 9th ACM Conference on Recommender Systems (RecSys 2015), Poster Session. Vienna, Austria, September 2015.

X. Amatriain and J. M. Pujol. Data Mining Methods for Recommender Systems. In Recommender Systems Handbook, pp. 227–262. Springer, 2015.

R. Baeza and B. Ribeiro. Modern Information Retrieval: The Concepts and Technology behind Search, 2nd Edition. ACM Press Books, 2011.

A. Barvinok. On the number of matrices and a random matrix with prescribed row and column sums and 0–1 entries. Advances in Mathematics 224(1), May 2010, pp. 316-339.

A. R. Benson, R. Kumar and A. Tomkins. Modeling User Consumption Sequences. 25th International Conference on World Wide Web (WWW 2016). Montréal, Canada, April 2016, pp. 519-529.

A. Bellogín, P. Castells and I. Cantador. Precision-Based Evaluation of Recommender Systems: An Algorithmic Comparison. 5th ACM Conference on Recommender Systems (RecSys 2011). Chicago, Illinois, October 2011, pp. 333-336.

Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager and A. Mahanti. Characterizing and Modeling Popularity of User-generated Videos. 29th IFIP WG 7.3 International Symposium on Computer Performance, Modeling, Measurements and Evaluation 2011 (IFIP Performance 2011) 68(11). Amsterdam, Netherlands, October 2011, pp. 1037-1055.

E. R. Canfield, C. Greenhill and B. D. McKay. Asymptotic enumeration of dense 0–1 matrices with specified line sums. Journal of Combinatorial Theory, Series A 115(1), January 2008, pp. 32-66.

R. Cañamares and P. Castells. Exploring Social Network Effects on Popularity Biases in Recommender Systems. 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) at the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, CA, USA, October 2014.

O. Celma. Music Recommendation. In Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space, pp.43-85. Springer-Verlag Berlin Heidelberg 2010.

O. Celma and P. Herrera. A new approach to evaluating novel recommendations. 2nd ACM Conference on Recommender Systems (RecSys 2008). Lousanne, Switzerland, October 2008 pp. 179-186.

Z. Cheng and N. Hurley. Effective diverse and obfuscated attacks on model-based recommender systems. 3rd ACM Conference on Recommender Systems (RecSys 2009). New York, NY, USA, October 2009 pp 141-148.

P. Cremonesi, Y. Koren and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. 4th ACM Conference on Recommender Systems (RecSys 2010). Barcelona, Spain, September 2010 pp. 39-46

P. Cremonesi, F. Garzotto, R. Pagano and M. Quadrana. Recommending without short head. 23rd International Conference on World Wide Web (WWW 2014 Companion Volume). Seoul, Republic of Korea, April 2011 pp. 245-246.

S. Goel, A. Broder, E. Gabrilovich and B. Pang. Anatomy of the long tail: ordinary people with extraordinary tastes. 3th ACM International Conference on Web Search and Data Mining (WSDM 2010). New York, NY, USA, February 2010, pp. 201-210.

B. Golshan, J. W. Byers and E. Terzi. What do row and column marginals reveal about your dataset? 27th Conference on Neural Information Processing Systems (NIPS 2013). Lake Tahoe, USA, December 2013, Pp, 2166-2174.

C. Greenhill, B. D. McKay and X. Wang. Asymptotic enumeration of sparse 0-1 matrices with irregular row and column sums. Journal of Combinatorial Theory, Series A 113(2), February 2006, pp. 291-324.

F. M. Harper, X. Li, Y. Chen and J. A. Konstan. An Economic Model of User Rating in an Online Recommender System. 10th International Conference on User Modeling (UM 2005). Edinburgh, Scotland, UK, July 2005, pp. 307-316.

E. Hensinger, I. Flaounas and N. Cristianini. Modelling and predicting news popularity. Pattern Analysis & Applications 16(4), November 2013, pp. 623-635.

J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 22(1), January 2004, pp. 5–53.

Y. Hu, Y. Koren and C. Volinsky. Collaborative filtering for implicit feedback datasets. 8th IEEE International Conference on Data Mining (ICDM 2008). Pisa, Italy, December 2008, pp. 263-272.

S. Krishnan, J. Patel, M.J. Franklin and K. Goldberg. Social Influence Bias in Recommender Systems: A Methodology for Learning, Analyzing, and Mitigating Bias in Ratings. 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, Silicon Valley, USA, October 2014, pp. 137-144.

K. Lee and K. Lee. My head is your tail: applying link analysis on longtailed music listening behavior for music recommendation. 5th ACM Conference on Recommender Systems (RecSys 2011). Chicago, Illinois, October 2011, pp. 213-220.

G. Linden, B. Smith and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1), January 2003, pp. 76-80.

B. Marlin and R. Zemel. Collaborative prediction and ránking with non-random missing data. 3rd ACM Conference on Recommender Systems (RecSys 2009). New York City, NY, USA, October 2009, pp. 5–12.

B. Marlin, R. Zemel, S. Roweis, and M. Slaney. Collaborative filtering and the missing at random assumption. 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007). Vancouver, BC Canada, July 2007, pp. 267–75.

A. N. Meltzoff and W. Prinz. The imitative mind: Development, evolution, and brain bases, 1st Edition. Cambridge University Press, 2002.

N. E. Miller and J. Dollard. Social Learning and Imitation, New edition. Greenwood Press Reprint, 1979.

M. Nakatsuji, Y. Fujiwara and A. Tanaka. Classical Music for Rock Fans?: Novel Recommendations for Expanding User Interests. 19th ACM International Conference on Information and Knowledge Management (CIKM 2010). Toronto, Canada, October 2010, pp. 949-958.

M. E. J. Newman. Networks, an Introduction, 1st Edition. Oxford University Press, 2010.

X. Ning, C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendation methods. In Recommender Systems Handbook, pp. 37-76. Springer, 2015.

J. Oh, S. Park, H. Yu, M. Song and S. T. Park. Novel Recommendation Based on Personal Popularity Tendency. IEEE 11th International Conference on Data Mining (ICDM 2011). Vancouver, Canada, December 2011, pp. 507-516.

K. Onuma, H. Tong and C. Faloutsos. TANGENT: A Novel, "Surprise-me", Recommendation Algorithm. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009). Paris, France, June 2009, pp. 657-666.

B. Pradel, N. Usunier and P. Gallinari. Ránking with nonrandom missing ratings: influence of popularity and positivity on evaluation metrics. 6th ACM Conference on Recommender Systems (RecSys 2012). Dublin, Ireland, September 2012, pp. 147–154.

J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer and A. Vespignani. Characterizing and Modeling the Dynamics of Online Popularity. Physical Review Letters 105(15), October 2010.

F. Ricci, L. Rokach and B. Shapira (Eds.). Recommender Systems Handbook, 2nd Edition. Springer, 2015.

M. J. Salganik, P. S. Dodds and D. J. Watts. Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. Science 311(5762), February 2006, pp.854-856.

A. Said and A. Bellogín. Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. 8th ACM Conference on Recommender Systems (RecSys 2014). San Jose, CA, USA, October 2014, pp. 129-136.

G. Shani and A. Gunawardana. Evaluating recommendation systems. In Recommender Systems Handbook, pp. 265–308. Springer, 2015.

A. Sharma, J. M. Hofman and D. J. Watts. Estimating the Causal Impact of Recommendation Systems from Observational Data. 16th ACM Conference on Economics and Computation (EC 2015). Portlan, Oregon, June 2015, pp, 453-470.

H. Shen, D. Wang, C. Song and A. L Barabási. Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes. 28th AAAI Conference on Artificial Intelligence (AAAI 2014). Hilton, Québec, July 2014, pp. 291-297.

H. Steck. Evaluation of recommendations: rating-prediction and ránking. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (RecSys 2013). Hong Kong, October, 2013, pp. 213-220

H. Steck. Item popularity and recommendation accuracy. 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (RecSys 2011). Chicago, IL, USA, October 2011, pp. 125-132.

H. Steck. Training and testing of recommender systems on data missing not at random. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010). Washington, DC, USA, July 2010, pp. 713-722.

J. Su, A. Sharma and S. Goel. The Effect of Recommendations on Network Structure. 25th International Conference on World Wide Web (WWW 2016). Montréal, Canada, April 2016, pp. 1157-1167.

G. Szabo and B. A. Huberman. Predicting the Popularity of Online Content. Communications of the ACM 53(8), August 2010, pp. 80-88.

W. Trotter. The Instincts of the Herd in Peace and War, 1st Edition. London Unwin, 1916. (A new edition was published by Cosimo Classics in 2005).

T. Wang and D. Wang. Why Amazon's Ratings Might Mislead You: The Story of Herding Effects. Big Data 2(4), December 2014, pp.196-204.

X. Zhao, Z. Niu and W.Chen. Opinion-Based Collaborative Filtering to Solve Popularity Bias in Recommender Systems. 24th International Conference on Database and Expert Systems Applications (DEXA 2013). Prague, Czech Republic, August 2013, pp. 426-433.

P. Zhang, M. Li, L. Gao, Y. Fan and Z. Di. Characterizing and Modeling the Dynamics of Activity and Popularity. PlosONE 9(2), February 2014.

Anexo 1: Demostraciones

Lema 1:

Cualquier ránking puede generarse a partir del ránking ordenado en sentido decreciente por $g(i)$, donde g es una función del ítem i, mediante una combinación de transposiciones de ítem adyacentes i_l, i_{l+1} donde $g(i_l) > g(i_{l+1})$.

Demostración:

Sea n el número total de ítems y sea R_n el ránking producido al ordenar dichos ítems en orden decreciente de $g(i)$.

Demostrar el lema es equivalente a demostrar que R_n puede obtenerse a partir de cualquier ránking mediante una combinación de transposiciones de ítem adyacentes i_l, i_{l+1} donde $g(i_l) < g(i_{l+1})$.

Para la demostración procedemos por inducción en el número de ítems n:

- Caso base: $n = 2$.

Con dos ítems únicamente hay dos posibles ránkings – i_1, i_2 y i_2, i_1 – y para pasar de uno a otro únicamente es necesaria una transposición adyacente, por lo que la demostración es trivial.

- Caso n implica el caso $n + 1$

Hipótesis de inducción: Dados n ítems, R_n puede obtenerse a partir de cualquier ránking mediante una combinación de transposiciones de ítem adyacentes i_l, i_{l+1} donde $g(i_l) < g(i_{l+1})$.

Sea un ránking genérico de $n + 1$ ítems

$$R = i_1, \dots i_n, i_{n+1}$$

Consideremos el subránking $i_1, \dots i_n$. Está formado por n ítems, luego podemos aplicar la hipótesis de inducción y deducir que a partir de él se puede generar R_n mediante transposiciones de ítems adyacentes i_l, i_{l+1} donde $g(i_l) < g(i_{l+1})$.

Mediante dichas transposiciones aplicadas al ránking R obtendríamos el ránking $R' = R_n, i_{n+1}$. Se trata, por tanto, de llevar el ítem i_{n+1} a su posición correcta dentro del ránking R_n, que recordamos está ordenado. Claramente i_{n+1} se puede colocar en su lugar intercambiándolo con los ítems que hay anteriores a él y que presentan un valor menor de $g(i)$. Cuando el siguiente tenga un valor mayor, esa es su posición.

Lema 2 (del orden óptimo):

El valor óptimo en cuanto a precisión observada se obtiene ordenando los ítems por el cociente $\bar{C}(i)$, mientras que para la precisión real se deben ordenar por $C(i)$, presentando dichos cocientes las siguientes expresiones:

$$\bar{C}(i) = \frac{p(seen|\, rel, i)\; p(rel|i)}{1 - \rho\; p(rate|seen, i)\; p(seen|\, i)}$$
$$C(i) = \frac{p(rel|i)\big(1 - \rho\; p(rate|seen, rel)\; p(seen|\, rel, i)\big)}{1 - \rho\; p(rate|seen, i)\; p(seen|\, i)}$$

Demostración:

Empecemos en primer lugar por la precisión observada, cuya expresión viene dada por la siguiente fórmula (deducida en el capítulo 4):

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} p(test, rel|i_k, R) \prod_{j=1}^{k-1} p\big(training\big|i_j, R\big)$$

Conociendo el lema anterior, sólo necesitamos probar que, dado un ránking $i_1, \dots i_l, i_{l+1}, \dots, i_n$ con $|i_l| > |i_{l+1}|$, intercambiar los ítems i_l y i_{l+1} produce un ránking con una menor precisión observada.

De acuerdo con la fórmula anterior, la precisión en ambos ránkings es la siguiente:

$$\mathbb{E}[\bar{P}@1|\, i_1, \dots i_k, i_{k+1}, \dots\, , i_n] = C_1 + p(test, rel|i_k)C_2 + p(test, rel|i_{k+1})p(training|\, i_k)C_2 + C_3$$
$$\mathbb{E}[\bar{P}@1|\, i_1, \dots i_{k+1}, i_k, \dots\, , i_n] = C_1 + p(test, rel|i_{k+1})C_2 + p(test, rel|i_k)\; p(training|i_{k+1})C_2 + C_3$$

Donde C_1, C_2 y C_3 son constantes en el sentido en que no dependen de i_l o i_{l+1} y, por tanto, no cambian al intercambiar ambos ítems[19].

La diferencia entre las precisiones de ambos ránkings es por tanto:

$$C_2\Big(p(test, rel|i_k)\,\big(1 - p(training|i_{k+1})\big) - p(test, rel|i_{k+1}, R)\big(1 - p(training|\, i_k)\big)\Big) = C_2\big(p(test, rel|i_k)\; p(\neg training|i_{k+1}) - p(test, rel|i_{k+1}, R)p(\neg training|\, i_k)\big)$$

Queremos que dicha diferencia sea positivo, es decir, que

$$\frac{p(test, rel|i_k)}{p(\neg training|\, i_k)} > \frac{p(test, rel|i_{k+1})}{p(\neg training|\, i_{k+1})}$$

Por tanto, hay que ordenar por

[19] Los valores de las tres constantes son:

$$C_1 = \sum_{k=1}^{l-1} \prod_{j=1}^{k} p\big(training|i_j, R\big)\;\; C_2 = \prod_{j=1}^{l-1} p\big(training|i_j, R\big)\;\; C_3 = \sum_{k=l+1}^{n} \prod_{j=1}^{k} p\big(training|i_j, R\big)$$

$$\frac{p(test, rel|i)}{p(\neg training|\ i)}$$

Descomponiendo $test$ y $training$ en función de $rate$, $seen$ y ρ se tiene que:

$$\begin{aligned} p(\neg training|\ i) &= 1 - p(training|\ i) = 1 - \rho\ p(rate|\ i) \\ &= 1 - \rho\ p(rate|seen)p(seen|\ i) \\ p(test, rel|i) &= p(test|\ rel, i)\ p(rel|i) = (1-\rho)\ p(rate|\ rel, i)\ p(rel|i) \\ &= (1-\rho)\ p(rate|seen, rel)\ p(seen|\ rel, i)\ p(rel|i) \end{aligned}$$

Y por tanto el orden óptimo para la precisión observada es por:

$$\bar{C}(i) = \frac{(1-\rho)\ p(rate|seen, rel)\ p(seen|\ rel, i)\ p(rel|i)}{1 - \rho\ p(rate|seen)\ p(seen|\ i)}$$

Procedemos análogamente para la precisión real, cuya fórmula tiene la siguiente expresión:

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} p(\neg training, rel|i_k, R) \prod_{j=1}^{k-1} p\big(training\big|i_j, R\big)$$

Observamos que la forma es muy similar a la precisión observada, únicamente es necesario sustituir $test$ por $\neg training$, de donde se deduce que el orden óptimo viene dado por el cociente:

$$\frac{p(\neg training, rel|i)}{p(\neg training|\ i)}$$

Que al introducir $rate$, $seen$ y ρ resulta en:

$$C(i) = \frac{p(rel|i)\big(1 - \rho\ p(rate|seen, rel)\ p(seen|\ rel, i)\big)}{1 - \rho\ p(rate|seen)\ p(seen|\ i)}$$

Anexo 2: Otras métricas

Evaluación mediante las métricas precisión, MRR, recall y nDCG sobre las 10 primeras posiciones de los ránkings, al tomar como entrada los datos de Crowdflower.

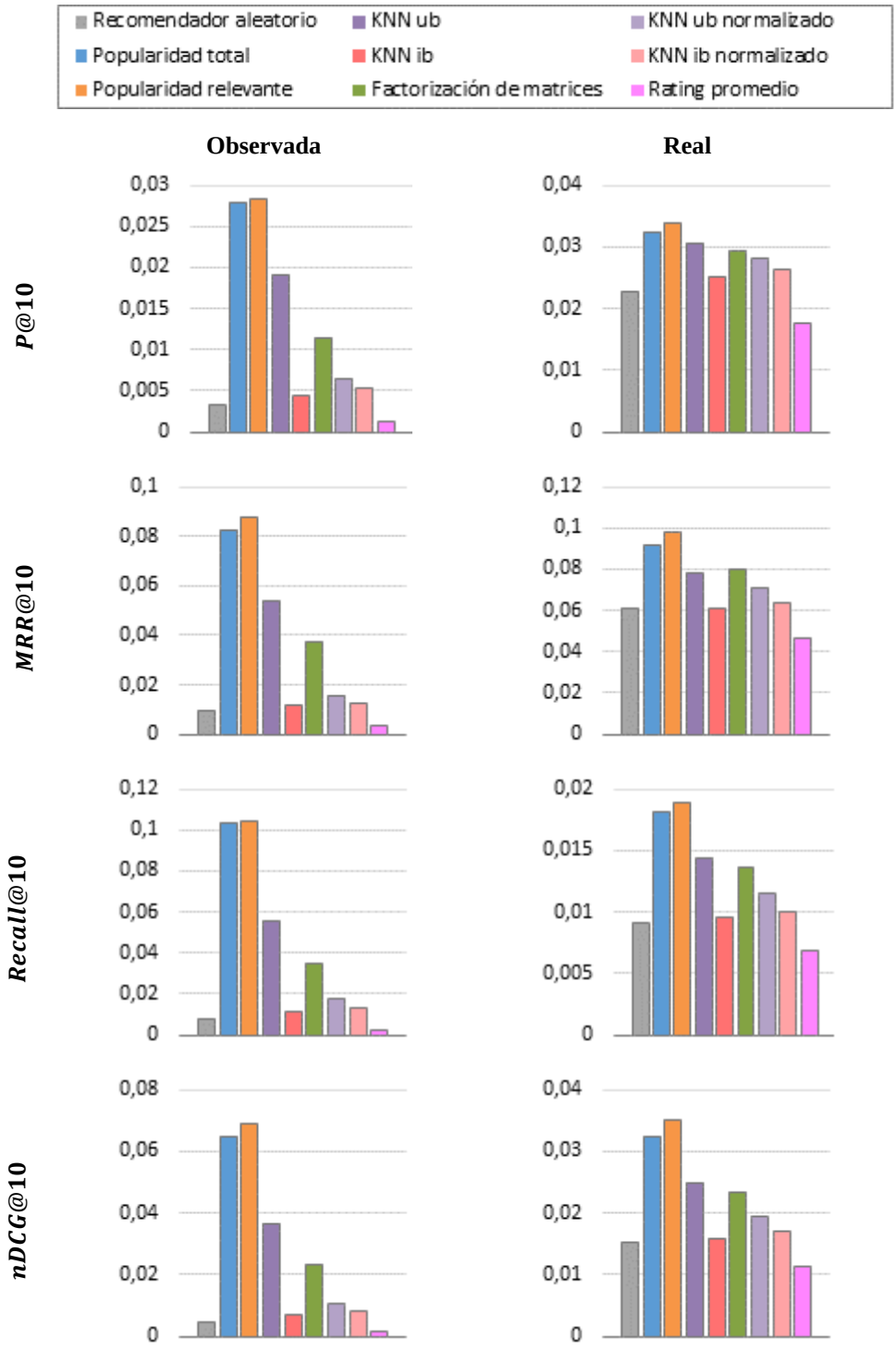

Figura 38. Valor observado (columna izquierda) y real (columna derecha) de las métricas precisión, MRR, recall y nDCG al evaluar las 10 primeras posiciones de los ránkings producidos por diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower, con una tasa de entrenamiento 0.5.

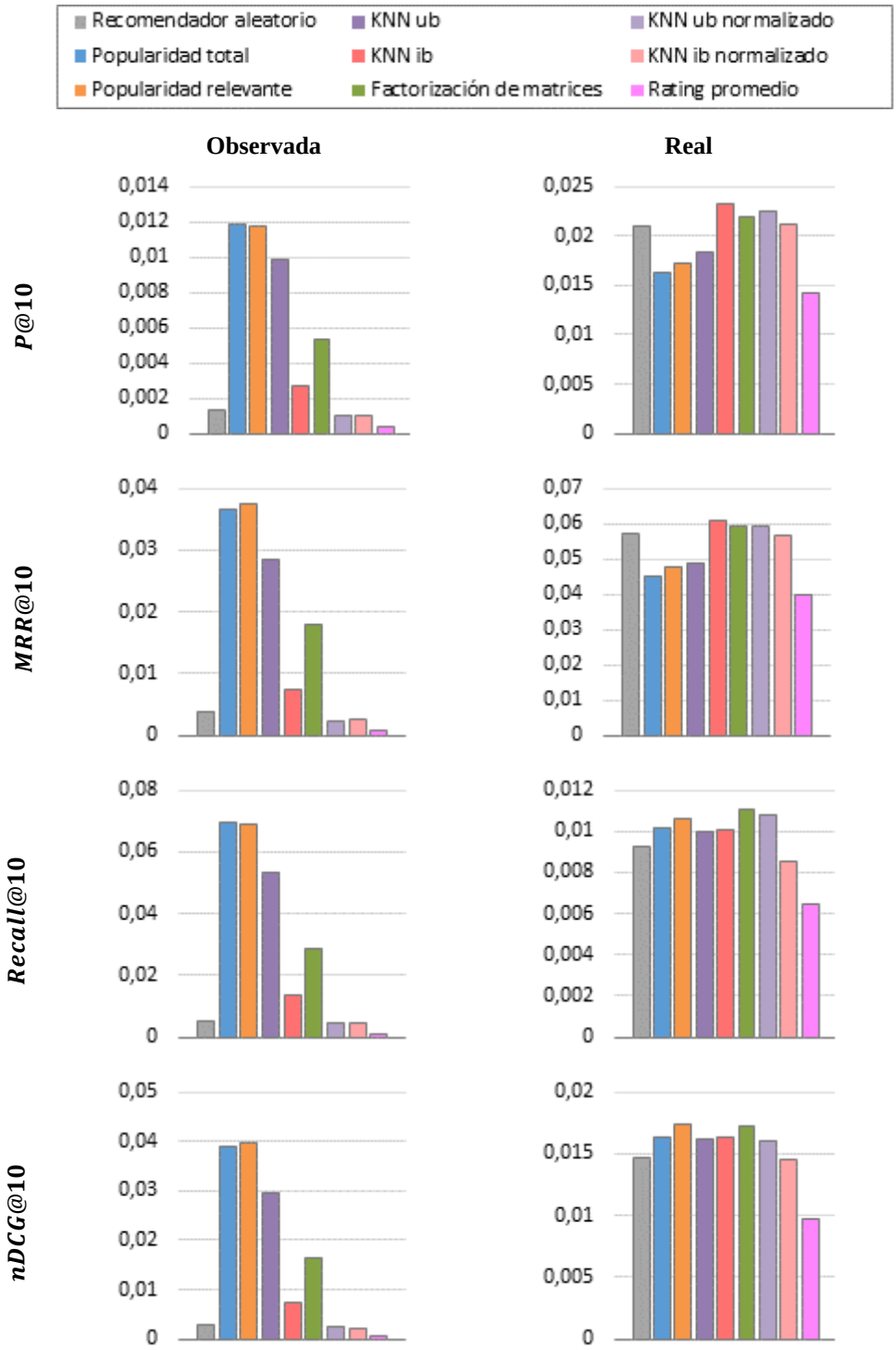

Figura 39. Valor observado (columna izquierda) y real (columna derecha) de las métricas precisión, MRR, recall y nDCG al evaluar las 10 primeras posiciones de los ránkings producidos por diversos recomendadores al tomar como datos de entrada las preferencias de los usuarios de Crowdflower, con una tasa de entrenamiento 0.8.

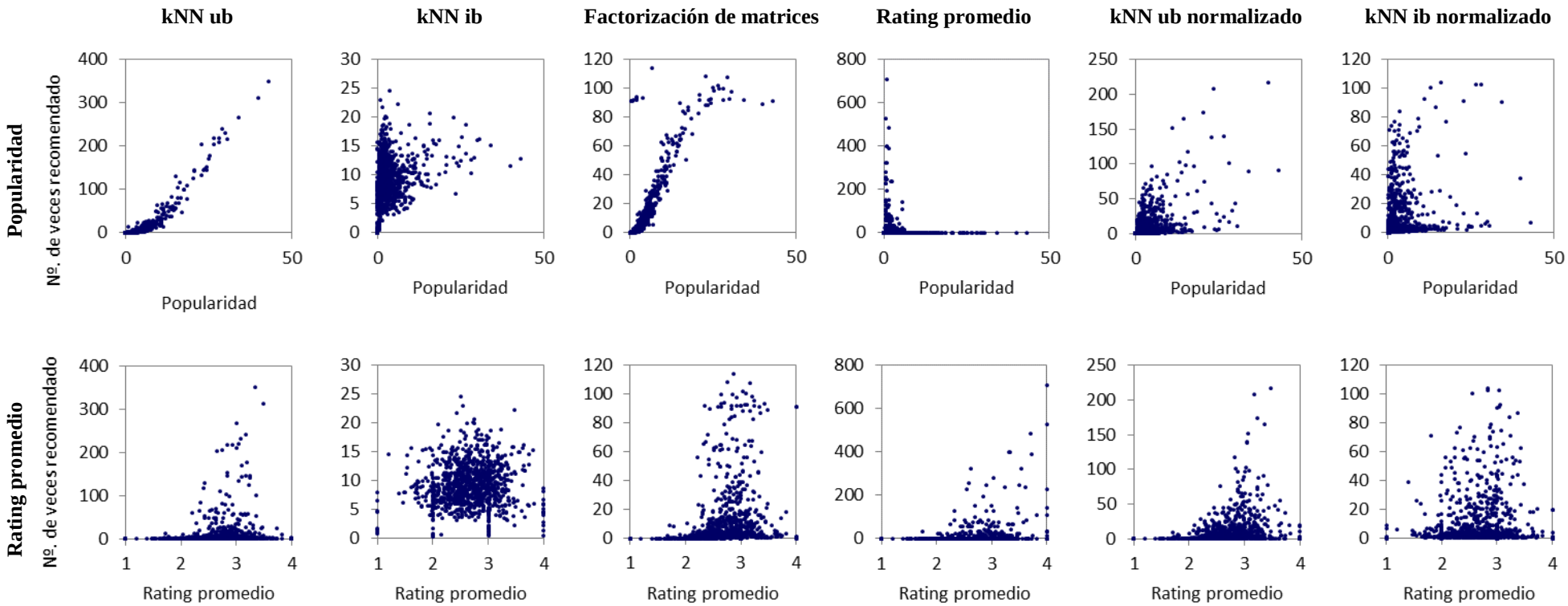

Figura 40. Número de veces que se recomienda cada ítem frente a la popularidad (relevante) – fila superior – que presenta dicho ítem y frente a su rating promedio – fila inferior – para diversos recomendadores. Los ránkings son de 10 elementos y la tasa de entrenamiento es 05.

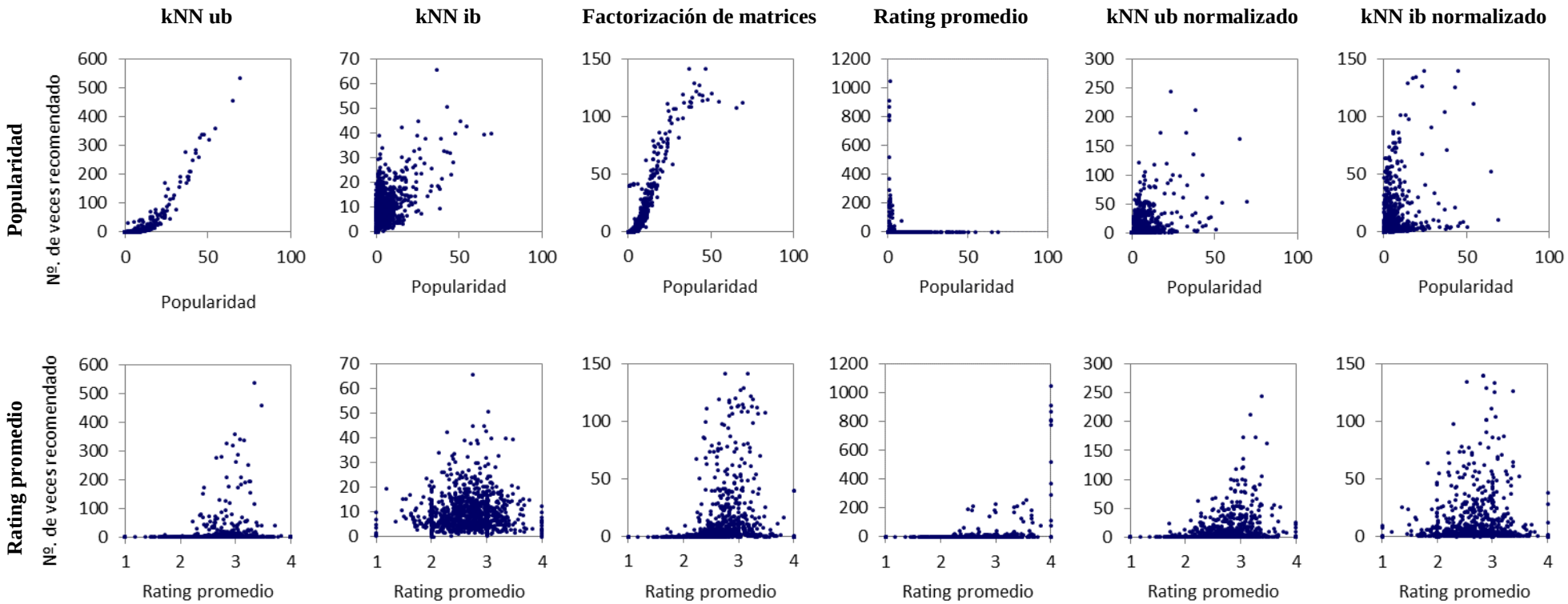

Figura 41. Número de veces que se recomienda cada ítem frente a la popularidad (relevante) – fila superior – que presenta dicho ítem y frente a su rating promedio – fila inferior – para diversos recomendadores. Los ránkings son de 10 elementos y la tasa de entrenamiento es 0.8.